EEG-Based Evaluation of Cognitive and Emotional Arousal when Coding in Different Programming Languages
Abstract

Cognitive psychology is a study of the brain, an organ that behaves as a complex computing system. The brain signals generate electrical signals, which can be interpreted meaningfully in line with the actions performed by the brain using various computational devices and measures using electroencephalography methodology. In this thesis, the signals obtained from the brain are processed to quantitatively study and compare the brain activities of coders while programming in two different programming languages. In this research, we have chosen the structured programming language C and the scripting language Python for comparison. Previous empirical research comparing various programming languages in a controlled manner identified attributes such as correctness, robustness, syntax, efficiency, etc. as parameters that characterize those programming languages (see e.g., Nanz and Furia, 2015; Garcia Jarvi, Lumsdaine, Siek and Willcock, 2003). This thesis aims to build upon the previous findings and compare the psychological effects during programming tasks.

Emotiv Epoc is a Brain Computer Interface device used for reading and analyzing brain signals in this study. Understanding the usage of the Emotiv device and the corresponding software tools is an essential part of this thesis work. Thus, in this thesis a pilot study is planned and a controlled experiment is conducted in order to collect and evaluate the data collected from the Emotiv Epoc device and self-reports to derive meaningful statistical results and interpret the emotional and cognitive activity of the participants.

The pilot study aims to understand how emotions and/or cognitive load vary while coding in C and Python. Initial study involves understanding the EEG method, principles and complexities involved with the collection of data. The core part of the thesis consists of planning and conducting a lab experiment, which involves six participants performing predefined tasks in C and Python, and answering a series of questionnaires. The collected data is analyzed in SPSS tool based on factors like time, performance (correctness), and questionnaire-based self-reports. EEG signals are analyzed in this thesis up to the point of artifact removal (Filtering and ICA using MATLAB and EEGLAB).

The results of the thesis reveal findings from emotional and EEG data analysis which have been captured from an experimental setup using the Emotiv device, software developers as participants and software programming tasks. The thesis helps to understand the steps to perform such an experimental study and to assimilate the emotional and cognitive load that affects the brain when performing programming. The particular comparison of C and Python as programming languages shows the high correlation between the programming language characteristics (such as syntax and time to code) and emotional and cognitive behavior of the programmers.

Keywords
Emotiv Epoc, quantitative, programming languages, EEG, emotions, cognitive load.

Supervisor
Dr. Dorina Rajanen
Foreword

First and foremost, I would like to thank Dr. Dorina Rajanen for giving me the opportunity to write my Master thesis in the Software Engineering for the chair of Information Processing Science at University of Oulu. This thesis topic was proposed by Dr. Dorina Rajanen. I am very grateful for her valuable inputs, collaboration efforts and immense support that contributed to the successful completion of the thesis work. I wholeheartedly thank my mentor for her persistent motivation and the extensive advice and support throughout the course of this research work.

I would also like to thank Dr. Mikko Rajanen for supporting me on all technical issues and important feedback during research.

A special thanks to Lic.Phil. Jouni Lappalainen for providing a valuable feedback and reviewing the final thesis work.

The thesis work was supported by the EMSE (European master of software Engineering) coordinators Prof Dr. Burak Turhan (University of Oulu, Finland) and Prof Dr. Dieter Rombach (Technical University Kaiserslautern, Germany), as well as Christian Wolschke, my EMSE coordinator at TUKL, Germany.

I would like to epically thank for all the participants who spent their valuable time and contributed to the experiment.

I want to specially thank all my friends who were directly or indirectly involved during the course of this thesis and took time to help me out and motivate me in my difficult times. I would like to specially thank my wife Ankitaa Bhowmick for her valuable inputs on technical issues and constant support. Also, I would like to extend my special token of thanks to my special friends Aniqa Rehman, Ayush Verma, Aprilia Hartami, Leevi Rantala and Samitha Jayathilaka; for being my strongest pillars of support. Last but not least, I thank my parents for always supporting and backing me up in all my endeavors.

Amit Rajendra Desai

Oulu, May 29, 2017
Contents

1. Introduction 6
   1.1 Background and Motivation 6
   1.2 Research Aim and Research Questions 7

2. Related Work 9
   2.1 Psychophysiological research using electroencephalography 9
      2.1.1 Electroencephalography (EEG) 10
      2.1.2 Principles of EEG measurement 10
      2.1.3 Measures obtained using EEG 11
      2.1.4 Applications of EEG in IS/SE research 14
   2.2 Emotiv Device 15
      2.2.1 The design, features, and usage of BCI devices 16
      2.2.2 Emotiv Epoc as a wireless EEG tool and BCI device: Methodology and characteristics 18
      2.2.3 Applications of Emotiv Epoc in Software Engineering 18

3. Research methods 21
   3.1 Questionnaire design 22
   3.2 Measures 22
      3.2.1 EEG based measures 22
      3.2.2 Self-report measures 23
         Background information 23
         Questionnaire in Programming 24
         Overall experiment 24
   3.3 Planning the Experiment 25
   3.4 Data analysis 25

4. Implementation 27
   4.1 Experimental Design 27
   4.2 Materials and Equipment 29
   4.3 Questionnaire Data 31
      4.3.1 Background information questionnaire 31
      4.3.2 Programming Task Questionnaire 31
      4.3.3 Overall Experience Questionnaire 32
   4.4 EEG recording 32
   4.5 Data Collection Problems 33
   4.6 EEG Data Analysis 35

5. Results 39
   5.1 Participants 39
5.2 Experimental Task Analysis 42
5.3 Overall Experience Analysis 45
6. Discussion 49
   6.1 Answers to the Research Questions 49
   6.2 Implication of the Research 50
7. Conclusion 52
8. References 53
   APPENDIX A. Consent Form for Participants 59
   APPENDIX B. C and Python Programming Tasks 60
   APPENDIX C. Background Questionnaire 63
   APPENDIX D. Programming Questionnaire 67
   APPENDIX E. Overall Experience Questionnaire 68
   APPENDIX F. Checklist 69
1. Introduction

Humans are often very emotional; our brain generates signals for every small activity performed in our daily life, whether playing games on computer, watching movies or a play, interacting with computers, or even programming. Emotions are feelings derived from the current state of mind and these emotions can be recognized via speech, facial expressions, textual information or even a gesture. Emotions can be categorized as positive or negative emotional states whereas cognitive load is the amount of work that the brain puts in to perform certain activity. (Liu, Sourina & Nguyen, 2010; Ahmad et al., 2016.)

The emotions of programmers during programming is an interesting study of human emotional states and cognitive load. To validate the emotions while programming, it is important to understand the programmer’s psychology and the amount of effort put into completing a programming task. Programming languages are complex. They can be categorized as simple as long as there is no debugging involved and can get very complex when an error is encountered. (Hoc, 2014.) Thus, a comparative study of programmer’s emotional states while programming in different programming languages gives an insight to the correlation between programming language complexity and the corresponding psychology of the programmers while programming in that particular language.

The section 1.1 provides an overview of the research topic and motivation to pursue the study. The aim and scope of the study is set in section 1.2.

1.1 Background and Motivation

In software engineering, we study several courses related to programming languages. A software engineer is intended to learn and use several programming languages. Programming languages can be categorized as structured, object-oriented, scripting, etc. Lahtinen, Ala-Mutka & Järvinen (2005) conducted a survey of novice programmers, which indicated that C++ was more difficult than Java. The survey results also stated that programmers were finding it more difficult to understand the basic concepts of programming rather than the application of the logic. Software developers’ emotions can be identified based on the progress of the programming tasks. If the positive emotions like happiness is observed one can assume that the task is in progress and if the negative emotions like frustration is observed, then it would seem that he is stuck in the task. (Muller & Fritz, 2015.) This thesis aims to study and understand the differences in programming languages based on the emotional and cognitive activities of programmers while programming.

For this purpose, a pilot study comparing two programming languages is conducted, in order to understand the programmer’s emotions and cognitive load while performing the programming tasks. To limit the scope of this study, two widely-used programming languages are selected; C which is a low-level structured programming language and Python which is a higher-level scripting language with simpler syntactical constructs as compared to languages. Nanz and Furia (2015) made the comparison of programming languages in order to find out the differences between procedural, object-oriented, functional, and scripting languages. Some of the factors considered for pairwise comparison in that experiment were lines of code, compile time, run time, patch and merge. The various programming languages that were compared in this experiment were C, C#, Java, Python, Ruby, Go, etc.
The results of the comparative study showed a significant difference between C and Python for the pairwise comparison mentioned above. (Nanz and Furia, 2015.) Hence C and Python languages were selected for the comparative pilot study in this thesis.

The motivation of this pilot study is to gain an in-depth understanding of the developers’ emotions and cognitive dimensions while they code. This analysis will help the software engineering community to understand the challenges like comprehension, attention, focus and memory load that may be encountered while programming. This research, which makes a comparative study, can provide valuable information to enhance the learning capabilities and skills of a programmer. Going further, in order to generalize this research a considerable number of participant from different backgrounds must participate in the research experiment. This would also eventually improve the results of the comparative study. The Emotiv Epoc device used in this research is a low cost commercially available device, but for further studies any other EEG device could be used. The steps formulated for analysis of EEG signals provides the basis for future application development using Emotiv Epoc device.

1.2 Research Aim and Research Questions

The study of psychophysiological research combined with software engineering has an extended scope of study. The aim of this research is to examine the emotions and cognitive load of the developers during programming in two programming languages. In particular, this thesis aims to conduct an exploratory pilot experiment where EEG data as well as self-reports are collected in order to measure emotions and cognitive load during programming. The following research questions are addressed in this thesis.

RQ1. What are the steps to be followed while using Emotiv Epoc device starting from recording of brain signals to analysis of the data?

RQ2. What is the cognitive and emotional activity as measured by different EEG-based measures and self-report measures of the participants when programming in two of programming languages of software engineering (C and Python)?

RQ3. What differences are observed between programming in C and Python with respect to programmer's performance, emotions and cognitive load during programming?

To conduct this study an empirical approach is adopted and an experiment is conducted in order to collect and analyze the data of interest. The experiment for data collection is carried out in a controlled environment and the data collected is analyzed using descriptive statistics, Pearson correlation coefficient and pie and bar charts using SPSS tool. The research method used in this study is a quantitative approach that provides an evaluation of the emotional states and cognitive dimensions measured during pre-defined programming tasks. In quantitative research the analysis is carried out in an unbiased way to deal with data, facts and figures. It relies on numbers/numerical values which are obtained when a phenomenon of interest is measured. (Muijs, 2010 and Jedlitschka & Rombach, 2016.) Thus, quantitative approach has been chosen in this research, based on the fact that the experiment provides numerical data that can be statistically analyzed in order to answer the research questions.
This thesis has the following structure: Chapter 2 aims to understand the background of psychophysiological research and the existing literature that provides the basic understanding like EEG signals and its measures for this thesis. Section 2.2 explains the features of Emotiv Epoc device and the feasibility of the device for this research. Chapter 3 presents the research methodology adopted for this pilot study and the measures which can be obtained for emotional and cognitive dimensions. In chapter 4 the implementation of experiment is explained in detail. Chapter 5 presents the results obtained from the study. In chapter 6 the results are discussed with respect to the research questions and the last chapter 7 presents the finding of the research work and future work related to this study.
2. Related Work

Since the thesis is based on psychophysiological study and its application in the field of software engineering, it is important to understand the related measurement and analysis techniques like electroencephalography (EEG). This chapter provides background information on EEG principles, its current usage in emotional and cognitive study and its further application in software engineering studies. The related work chapter also aims to provide the details of data capturing and analysis techniques in psychophysiological software engineering applications, thereby underlining the features and efficiency of Emotiv Epoc for this research.

In the first section 2.1, the existing research has been studied to understand the different psychophysiological methods to obtain the emotions and cognitive measures using EEG. In order to conduct a EEG-based research, it is important to understand how the previous studies were performed in relation to software engineering applications. The EEG is captured via Emotiv Epoc which is a commercially available, low cost BCI device. The section 2.2 shows the various research applications of Emotiv device and helps to draw a conclusion that this device fits the purpose of this study.

2.1 Psychophysiological research using electroencephalography

Psychophysiological research is the study of physiology, which deals with the relationship between mental and physical phenomena of a person. The non-invasive recording of the psychophysiological responses primarily focuses on collection of electrical signals at the surface of the skin. There are distinct types of measures employed in psychophysiological research. (see e.g., Potter and Bolls, 2012; Hodges, 2010.) Accordingly, the heart rate is measured using the electrocardiogram (ECG). The measure of gastric mobility is measured using electrogastrogram, blood pressure is captured by finger pulse amplitude (FPA), and muscular tension can be studied by using the electromyography (EMG). Brainwaves are captured using electroencephalography (EEG). In this study, the focus is on measuring brain activity using EEG, and therefore the literature review details the background and applications of EEG.

According to Potter and Bolls (2012), the psychophysiological measurements involve three primary approaches, namely self-reports, readings of the psychophysiological signals using specialized equipment, and observations of the external behavior of the study participants. The common usage of psychophysiological research is to study the emotions and to understand the cognitive processes such as attention and memory. The recording in psychophysiology focuses on collection of electrical signals or “biopotentials”, which are generated at the surface of the skin. Since these signals are measured at the scalp in the case of EEG, the strength of the EEG signals is very low. Therefore, the EEG equipment also involves the use of amplifiers in order to detect and record the actual signals. There has been significant research in improving the signals obtained from brain and preprocessing the data using EEGLAB, machine learning algorithms or Fast Fourier Transform for signal measurements (Lee et al., 2015; Khushaba et al., 2013; Khushaba et al., 2012; Trivedi, 2013). In the following subsection 2.1.1, the emphasis is given on understanding the concepts of EEG.
2.1.1 Electroencephalography (EEG)

The existence of human race is due to the fact that we have emotions. The ability to feel, touch, sense and act based on the emotional factor makes our life complete. Emotions in humans have a special meaning to life and it exists every second. The emotions cannot be expressed completely or measured using a survey or self-reports and hence an in-depth analysis can be performed by a psychophysiological study to validate the positive and negative emotions and cognitive measures like comprehension and attention of humans. (Potter and Bolls, 2012.)

EEG is a short-form for electroencephalography, which measures the central nervous system activity evoked by brain electrical signals (Potter and Bolls, 2012). Hans Berger discovered this measurement technique in late 1920, which was the first of its kind to measure electrical signals from brain. The experiment was conducted by soaking two sponges in saline and connected to differential amplifier. Since then there is a significant improvement in ways of collecting the EEG data. The electrical signals which can be received from the scalp can be measured to a very low range of $10^{-6}$ V. (Potter and Bolls, 2012.)

The brain is divided into four regions, which can be used for the measurement purpose; the frontal, parietal, temporal, and occipital lobes. The human brain is a huge mass of interconnected neurons. The neurons provide immense amount of biopotentials, which can be recorded by the EEG signals. (Potter and Bolls, 2012.) Two major types of brain waves that can be measured using EEG are: spontaneous EEG (referred as continuous EEG), and evoked potentials (referred as event-related potentials). Most of the research is done on the spontaneous EEG in clinical context. The spontaneous EEG signals are equally relevant in the field of cognitive neuroscience, psychophysiology, etc. (see e.g., Müller-Putz, Riedl and Wriessnegger, 2015.)

The further section provides an insight of the measurement in EEG and electrode placement system. This contains general information of how to record the brainwaves and principles behind the measurement.

2.1.2 Principles of EEG measurement

In psychophysiological research, typically the recording of the EEG is done using a stretch-lycra cap on which a number of electrodes are mounted (e.g., 32, 64, 128, or even more; Harmon-Jones and Amodio, 2012). The electrodes are made of tin or silver and silver chloride. With the cap, the electrodes are positioned over the entire scalp.

Typically, the placement of electrodes is done according to established standards such as the 10-20 system or 10-10 system (Harmon-Jones & Amodio, 2012; Oostenveld & Praamstra, 2001). The number ‘10’ and ‘20’ primarily refers to the distances between the electrode placement which is either 10% or 20% of the total front-back or right-left length. As shown in Figure 1, there are several points where the electrodes are placed in order to collect the brainwaves. The letter is indicated based on the position of the electrode placed. The letter FP means frontal pole, F stands for the frontal lobe, C means the central region, P is parietal lobe, T is the temporal lobe and O is the occipital lobe. (Alshbatat, Vial, Premaratne & Tran, 2014; Harmon-Jones & Amodio, 2012; Oostenveld & Praamstra, 2001.) The electrodes placed in between these regions are often represented with two letters and it measures the difference in voltage between the neurons at this region. The letters can be associated with numbers. The letters with odd numbers represent left side of the brain and the letter with even number refers to the right part of the brain. This representation helps to locate the designated site for measurement.
The numbers increase as the distance from the central(C) increases, like F3 is nearer to central and F7 is further from central. (Harmon-Jones & Amodio, 2012; Oostenveld & Praamstra, 2001.)

![EEG Electrode placement](image)

**Figure 1.** EEG Electrode placement (Müller-Putz et al., 2015).

The electrode reference points are marked with different colors as seen in Figure 1. The points circled in black represents the actual 10-20 system. The gray circles are the additional 10-10 extension system. To place the electrode for a designated site for measurement, the surface area of the scalp needs to be cleaned to reduce the electrode impedances that are mostly under 5000 ohms. At the scalp area, the recorded signals are sine wave with positive and negative deflection with a varying measure of frequency and amplitude. (Harmon-Jones & Amodio, 2012.)

### 2.1.3 Measures obtained using EEG

In this section, the different EEG measures are studied which are obtained from the EEG signals and its different frequency ranges. These measures help to understand the differences in emotions and cognitive load.

Emotions can be categorized as positive and negative emotional states which are expressed as a current state of the mind. The emotions are measured from the signals obtained from frontal asymmetry as explained later in this section. Cognitive dimensions are the amount of effort which are put to perform certain activity, e.g. attention, comprehension memory load, etc. These measures are obtained from different frequencies patterns. To understand the emotional and cognitive dimension it is important to know the different frequency bands. The study below indicates the different frequency bands like alpha, beta, gamma, theta and delta which gives the measures of emotional and cognitive dimensions.

In the psychology of emotion, there are several discrete feelings like anger, disgust, sadness, fear, shame and surprise (see Petrantonakis & Hadjileontiadis, 2010; and Ekman, 1992). Another conceptualization of emotions termed by Davidson, Schwartz, Saron, Bennett & Goleman (1979) views emotions as a two-dimensional state defined by valence (pleasantness or feeling of joy) and arousal (excitement or activation).
Valence is a judgement about a satiation if it is negative or positive whereas arousal is spanned from a range of calmness to excitement. (Petrantonakis & Hadjileontiadis, 2010.) Stress is a negative emotional experience which is an affect caused by external pressure to perform better or complete tasks in time (Hamid, Sulaiman, Aris, Murat & Taib, 2010). In some research papers the stress is assumed as two factors eustress and distress. Eustress relates to good stress such as joy and distress is negative emotions (Healey & Picard, 2005). According to Reisman (1997), “stress is a body’s reaction with the release of Cortisol (stress hormone) due to physical, mental or emotional pressure”.

Hamid et al. (2010) conducted a study to evaluate the stress using the EEG signals and analyzing the alpha and beta frequency EEG signals were measured. Healey & Picard (2005) studied stress using EEG, EMG, skin conductance and respiratory measures when driver are riding cars in Boston area.

Typically, in detecting emotions and cognitive dimensions, the raw EEG signal obtained can be decomposed into five distinct frequency bandwidths as described by Potter & Bolls (2012), Müller-Putz et al. (2015) and Petrantonakis & Hadjileontiadis (2010).

1. Alpha: The bandwidth frequency ranging from 8-13 Hz with very large amplitude and is mostly associated with relaxation and readiness.
2. Beta: The range of frequency bandwidth is about 13-25 Hz or 13-30 Hz with a lower amplitude. This is mostly related to alertness, concentration and anxiety.
3. Theta: It occurs at a bandwidth range of 5-7 Hz or 4-8 Hz and measures the cognitive dimensions like memory load, mediation, drowsiness and specific sleep states.
4. Delta: It is not a common research oriented measure in the field of science as it is mostly measured for deep sleep or coma at a bandwidth range of .5 to 4 Hz.
5. Gamma: This range of frequency is about 30-100 Hz which measures arousal and peak performance.

The EEG signal that is generated from the surface of the scalp is very low, in range of some microvolts. The signals need to be amplified many times in order to detect the original signal. (Potter & Bolls, 2012.)

The alpha band power is inversely correlated to the actual activity happening in the brain area (Tóth, 2015; Petrantonakis & Hadjileontiadis, 2011; Müller-Putz et al., 2015), while the power on all other frequency bands is directly correlated with the brain activity (higher brain activity is shown as higher power on theta, beta, and gamma frequency bands).

Frontal asymmetry is a measure of emotion based on brain signals (Coan & Allen, 2004; Davidson, 1993). To measure the frontal asymmetry, the frontal lobe is divided into two parts, left-frontal and right-frontal. The front-left region of brain activates during experiences of positive emotions like joy and happiness. The right-frontal area experiences the negative emotions like fear or disgust. If the alpha band power is high on the left hemisphere compared to the right part of the brain, it implies that the activity is high on the right hemisphere and vice versa. Using the 10-20 system to capture this information; the electrodes can be placed from left to right at frontal, central, temporal and parietal region (F3, F4, C3, C4, T3, T4, P3, P4). (Petrantonakis & Hadjileontiadis, 2010; Tóth, 2015; Schmidt & Trainor, 2001.)
Besides emotional states, EEG can be used to measure cognitive load. Haapalainen, Kim, Forlizzi, & Dey (2010) described cognitive load as “a multidimensional construct representing the load that a particular task imposes on the performer”. The mental work load is a measure of the interaction between the processing of the task and the capability one individual (Haapalainen et al., 2010 and Berka et al., 2004). Cognitive work load is the amount of work the brain inherits to achieve a task/goal (Anderson et al., 2011). There are several kinds of measures which can be obtained from the EEG signals that can indicate the level of comprehension, attention and memory load (Berka et al., 2004). These measures are derived by decomposing the EEG signals into frequency bands (e.g., the alpha, beta and theta bands) and calculating the power or amplitude of the oscillations in the specific frequency bands.

Comprehension is a measure obtained when a subject is in a state of understanding a particular stimuli or task. Lee et al. (2016) measured comprehension when participants were performing Java tasks by using the amplitude (power) of the beta and gamma frequency bands. The data was statistically analyzed based on the power of beta and gamma frequencies at frontal, parietal and central lobe by measuring the average mean at each lobe which checks for the activations. (Lee et al., 2016.)

Attention can be stated as the right amount of allocation of processing resources to focus to a stimulus and the term stimuli here can be considered as an object, location or moments in time. Attention can be divided into four subcategories or processes such as: (Coull, 1998.)

1. Attentional orientation: This kind of attention is observed when an attempt is made to complete a particular stimuli/goal,
2. Selective or focused attention: Prioritizing one task more over another,
3. Divided attention: Attention for more than one task at a time. In technical meaning, it can be termed as multitasking.
4. Sustained attention: Is to concentrate on an event or for a future task (Coull, 1998). Lee et al. (2016) measured sustained attention using the theta frequency band power.

The memory load is measured from the theta band frequency of 4-8Hz. The memory load patterns indicate the working memory during learning tasks (Jensen & Tesche, 2002; Onton, Delorme, & Makeig, 2005). Jensen & Tesche (2002) performed a study to indicate that the frontal theta activity increases the load on the memory while working on a memory-intensive task. The analysis was performed by comparing the Event-Related Potential (ERP) heat maps to indicate the brain activity during learning. Onton et al. 2005 performed a study to observe the memory load of participants while performing memorization tasks where a power spectra is used to analyze the activity in the frontal midline. These studies show that in order to obtain the memory load measures, it is important to observe the theta band frequency range.

In order to obtain the accurate frequency range from the EEG recording it is important to remove any kind of noise signals. The noise signals are termed as artifacts. The key issue in artifact removal process is the loss of good signals (Trivedi, 2013). Artifacts reduces the signal quality (Daly, Billinger, Scherer, & Müller-Putz, 2013). Hence it is very important to remove these artifacts carefully from the EEG signals. Artifacts can be caused due to movement of head, eye blink, voluntary muscle activity and noise from surrounding electronic activity. (Daly et al., 2013; Ranky & Adamovich, 2010.) Apart
from the artifacts the signal also contains spurious signals which needs to be removed (Nagy et al., 2014). The disturbance caused by the eye-blinks and eye movements are called as ocular artifacts. Hence the signals for the frontal portion might be irregularly distributed with high amplitude. The most common method employed to remove this artifact is by rejecting these artifacts when found. (Mantini et al., 2007.)

In an experiment conducted by Teplan (2002), it was found that the recording of EEG signals can have two types of artifacts; one is the patient-related artifact and another one is technical artifact. In a patient-related artifact, it is the unwanted signals which disturb the EEG signals, like sudden spikes. The technical related artifact can be caused by AC power line noise. (Teplan, 2002.) Hence different devices can cause different kind of artifact.

Another method to remove the artifacts is pre-processing of signals using filters to improve the signal to noise ratio. The types of filters usually used are the low-pass, high-pass, band-pass and band-stop filters. The high-pass filter is used when the force of the DC signals is reduced by applying the low frequency component on a direct-current signal. The low-pass filter is used when attenuating the high-frequency component to smoothen the filter output. Band-pass and band-stop filters combines the two low-pass and high-pass filter. (Widmann, Schröger, & Maess, 2015.)

2.1.4 Applications of EEG in IS/SE research

This section presents the related work regarding the research performed using different EEG devices in the field of software engineering. It is also important to understand the process and methodology in performing psychophysiological research.

Müller and Fritz (2015) conducted a study to investigate software developer’s emotions while performing a software change tasks. In this research, there were 17 participants who were given with two change tasks. Each task had a total duration of 30 minutes to complete the change. They wanted to measure the psychological measures like valence, arousal, happiness and sadness. To record the EEG signals they used Neurosky MindBand sensor device. To measure the skin and heart rate signals a wrist band from Empatcia E3 and Eye tracker: Eye tribe was used to collect the data. The goal of their study was to understand three factors. The first research goal was to understand the range of emotions which flow during the change tasks and if the developers’ emotions correlate with the progress of the task. The second research goal was to identify the practices which affect the developers’ emotions during the change task. The third one was to know if the biometric sensors could be used to determine the developers’ emotions during the change task. (Müller & Fritz, 2015.) In the experiment, participants were asked to first put on the Empatcia device to record the EEG signals. The device was checked for connectivity before the task and whether all reference points were in good health and in synchronization. Once the setup was completed they were provided instruction for task. Before starting each change task all participants were asked to watch a two-minute video of a fish swimming in a tank so that participants could relax and have a calm mind. To calculate the correlation of the perceived progress with emotions during the task all participants were interrupted every 5 minutes during each change task to rate the emotions if they felt happy or sad. To measure this value, a Likert scale of 5-point system was used. During the change task, all participants were given with internet access to perform a search. The task given for the participant was to implement an API to perform an interaction with StackExchange API and the other task was to create a new feature in open-source Java GUI framework. (Müller and Fritz, 2015.)
The results for the first research questions showed that valence was highly correlated with the perceived progress of task; The second result showed the participants requested for more time for completion of task and felt the pressure of time during completion. The third results indicate, the biometric device provides a very high accepting result to distinguish the positive and negative emotions. (Müller and Fritz, 2015.) Hence this result ensures the device can be used to determine the emotions.

Another study in software engineering using the EEG methodology was conducted to compare the programming comprehension and observe the differences between a novice and expert programmer using cognitive science (Lee et al., 2016). This experiment was performed with 18 participants to record the EEG signal and calculate the beta and gamma frequency to measure comprehension. The beta frequency measures the logical thinking and gamma frequency measures the attention and memory while processing information (Lee et al., 2016). The recording of EEG was made at 13 different sites on the scalp (F3, F4, F7, F8, Fz, C3, C4, Cz, P3, P4, P7, P8, Pz); in addition to the EMG and EOG were also recorded. The antiCAP by Brain products GmbH was used for EEG to capture the signals, while the analysis was done with Brain vision analyzer 2.1 which used a frequency filter to obtain the signals ranging between 4-50 Hz range. Capturing of activities like eye blinking, jaw clenching was done by EMG. Alpha, Beta, Theta and Gamma signals were captured to measure the concentration level while comprehension (Lee et al., 2016). The design of experiment was set up with three tasks. The first step was to complete the 36 Java questions and there was no time constrains to complete them. Each task was given with four multiple choice answers and questions were shuffled for every participant. The instruction given to participant was to close the eyes for 30 seconds after selecting the answer. Once the answers were chosen the brainwaves was recorded for 30 seconds for every question. In the step 3 the participants were woken up by an alarm which went off after 30 seconds and repeat the step 1. Apart from the experiment a survey was conducted prior to the experiment to understand if the participants were a novice or an expert the analysis of the results was that the novice users lacked the understanding of basics concepts, hence the experts have better comprehension in programming and solving simple tasks in brief time (Lee et al., 2016). The two-main findings in this research were the properties of brainwaves which are identified during comprehension, and the difference of brainwave pattern in the novice and expert users while performing the task.

The above research provides the methodological details of collecting EEG signals using different EEG-based devices. The steps like baseline recording, self-reports and analysis of EEG signals are studied in order to understand the process followed in software engineering research. Few ideas about the duration and activities performed during baseline recording are taken into consideration for this pilot study and accordingly steps have been formulated for the experiment. The background study also provides the various analysis methods best suited for EEG signals which help to formulate the analysis steps indicated in section 4.6.

2.2 Emotiv Device

The background study presents multiple devices and methodologies for capturing EEG signals and its analysis. For this thesis, Emotiv Epoc was chosen as the most feasible BCI device to record the brain signals. This section describes the features of Emotiv

---

1 https://www.emotiv.com/epoc/
Epoc and its existing research applications that prove its effectiveness for this study.

Emotiv Epoc is a wireless EEG measurement tool and Brain Computer Interface (BCI) device with 14 channels to record and measure the raw electroencephalography (EEG) signals. It is a wireless Bluetooth device that is compatible with Android, Linux, IOS and Windows operating systems. It is in the form of a headset device that is rechargeable. The device is very easy to use as it uses saline based wet sensors (no sticky gels). Since the device is a low cost commercially available tool the below study helps to understand the feasibility and usage of this device. The advantage of using this device is that the participant’s movement is not as constrained as compared to the biomedical device where the participant has to stay still with limited movement of head and body (Emotiv, 2016a; Alshbatat et al., 2014.)

2.2.1 The design, features, and usage of BCI devices

Emotiv Epoc device complies with many regulatory requirements like Federal communication commission (FCC) rules part 15, Radio standard specification (RSS)-210 and Low voltage requirement rule: Directive 2006/95/EC. FCC ensures that the device may not cause any harmful interference and the interference received should not cause any undesired operation. RSS-210 is to ensure that the device itself should not cause any interference. (Emotiv, 2016a.)

The software is open source, which allows users to develop applications based on different requirements. The device is compatible with different platforms and it can be used to develop mobile applications to provide live feedback to the users. The Emotiv has launched an Android application called MyEmotiv. This application helps us to track the brain’s fitness. It measures various cognitive and emotional measures like stress, excitement, focus, interest, engagement, and relaxation (Emotiv, 2016a).

The features of the Emotiv Epoc device is as listed in Table 1. The system should adhere to these specifications so that Emotive Epoc device operate as desired (Emotiv, 2016a). The minimum system specification required for the Emotiv device with Windows machine is Windows XP service 2 operating system and higher, 2 GB ram with a minimum space of 200 MB, USB port of 2.0 speed or higher, Intel Pentium 4 or equivalent processor (Emotiv, 2016a).

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Channels</td>
<td>14 biopotential sensors with gold plated connectors</td>
</tr>
<tr>
<td>Channels Names</td>
<td>AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4</td>
</tr>
<tr>
<td>Sampling Method</td>
<td>Sequential sampling. Single ADC</td>
</tr>
<tr>
<td>Sampling Rate</td>
<td>128 SPS or 256 SPS</td>
</tr>
<tr>
<td>Battery Life</td>
<td>Up to 12 hours using proprietary wireless, up to 6 hours using the device.</td>
</tr>
<tr>
<td>Battery Type</td>
<td>Internal Lithium Polymer battery 640mAh</td>
</tr>
<tr>
<td>Resolution</td>
<td>14 bits 1 LSB = 0.51μV</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>0.2 – 43Hz, digital notch filters at 50Hz and 60Hz</td>
</tr>
<tr>
<td>Coupling mode</td>
<td>AC Coupled</td>
</tr>
<tr>
<td>Connectivity</td>
<td>Wireless Bluetooth® Smart</td>
</tr>
</tbody>
</table>
Figure 2. Emotiv Epoc headset placement nodes (Emotiv, 2016a).

The Emotiv Epoc device has 14 channels for recording from the human scalp. The electrode sites are as shown in the Figure 2. The studies performed in section 2.2.2 indicates that this device can be used to measure the emotions and cognitive dimensions.

The Emotiv Epoc headset is a flexible headset which can be fitted to any person head regardless of size. The Figure 3 shows the Emotiv device which is used for this pilot study. It is wireless BCI device with simple and easy to wear device. Since this device is wireless it is useful to perform this research as the movement is not restricted and participant perform with ease.

Figure 3. Emotiv Epoc device (Emotiv, 2016a).

The device uses the Emotiv Control Panel to check if all the electrodes are connected to the scalp; for this purpose, the strength of the connection is indicated by green (perfect connection), orange, red and black (no connection) color for each electrode. The Emotiv Control Panel also provides an indicator of the battery. These indicators help to understand if the signals are captured with full strength. Emotiv Xavier Testbench is a tool to capture the EEG signals. This tool provides recording of EEG signals, setting baseline study, facial recognition signals recognition. The EEG signals captured are in European data format (EDF) that can be converted to comma separated value (CSV) format (Emotiv, 2016a.)
2.2.2 Emotiv Epoc as a wireless EEG tool and BCI device: Methodology and characteristics

Emotiv Epoc has been used by many researchers to perform different kinds of research in the field of science. In the research, Ranky & Adamovich (2010) provides information on how the Emotiv device is used to control the robotic arm and the adaptability it provides to multiple software programs, users and peripherals. The device requires minimal training time, no health risks, improved resolution and speed are the major advantage of measuring EEG using this device. Even though it requires filtering of noise signals, the efficacy of the Emotiv Epoc is proved by the robustness and cost effectiveness of the device. (Khushaba et al., 2012; Khushaba et al., 2013.)

Khushaba et al. (2013) performs a study to observe and evaluate the cortical activity of different brain regions and the interdependencies among EEG signals, which focuses on liking/disliking. Using the Emotiv device, the frontal (delta, alpha, gamma, theta and beta across the channels F3, F4, FC5, and FC6) were measured. The emotive Epoc frequency was set with 128 Hz per channel for measurement. The Emotiv Epoc device comes with a software development kit that provides the count on the packets transmitted and sent to ensure no data is lost. The data collected for the experiment were then analyzed using MATLAB. In addition to EEG measurements, the eye tracker device Tobii was also used to detect the images of the object. (Khushaba et al., 2013.) In this experiment 18 participants were recruited with an average age of 35 years. The average time for participants completing the task was 7 minutes which included reading instruction and 57 choice sets for completion. This task was to select least preferred cracker from each set of three objects (Khushaba et al., 2013). To analyze the signals obtained from the Emotiv Epoc, it was first required to remove the noise signals (Khushaba et al., 2013). They used Independent Component Analysis (ICA) and discrete wavelet transform from the data collected from Emotiv Epoc device. In the first part of the experiment, the aim was to calculate the phase locking value measure to determine the phase synchronization while participants chose the preferences for the different cracker characteristics. The finding supported the assumption made of having a wide inter-hemispheric communication during the experiment. In the second part of the experiment, the individual preferences for each cracker characteristics were analyzed and assessed. The Anova test results indicate that there was significant difference across the channels on the beta and gamma values. (Khushaba et al., 2013.)

In the research paper by Stamps and Hamam (2010) there was a study made to compare various BCI systems (Neurosky, Modular EEG, Neurobit, Psychlab, etc.) available for research. The results of the survey were based both on the hardware and software of the BCI system. Emotiv Epoc was ranked the best scoring 8.6 out of 10 in the survey results. The results were evaluated based on the hardware flexibility, commercial producibility, portability and ease of electrode placement. (Stamps and Hamam, 2010.)

2.2.3 Applications of Emotiv Epoc in Software Engineering

Emotiv Epoc gained recognition as being accurate and cost effective. Since the device comes with an open source software development kit, there are several applications in which the device is used in scientific and academic contexts. Its application is extended to gaming, education and training, defense, communication, automotive, and Internet of Things (IOT) development.
BCI device can be used to automate house appliances. Alshbatat et al. (2014) study is based on patients having problems with paralysis and cannot perform the day to day life activities. To enhance the quality of life for these patients the study was conducted to have a graphical user interface to control the home appliances with the help of Emotiv Epoc device. The Emokey application is used to map the thoughts of the patients to keyboard input. along with this tool the Emotiv Epoc provides a cognitive tool where the movement of cube can be trained and desired object of movement will be synced to perform the action. An Embedded model was created for this study purpose which could read the signals processed from the Emotiv software using a Wifi module. A Graphical user interface was developed to understand the key stroke and perform the action. This study provides us the extent to which this device can be used in scientific research and improve the quality of life. In this experiment one male participated. (Alshbatat et al., 2014.)

In the research paper (Duvinage et al., 2013), the Epoc performance is compared with the bio-medical field device to understand the performance and applicability of the device. The performance of the device was calculated on condition like walking on treadmill and sitting on a chair. The statistical comparison between the Advanced neuro technology (ANT) device with Emotiv Epoc device showed the device is quite suitable for gaming and not for medical grade system one. The device can also be used for rehabilitation system to improve the patient recovery process. (Duvinage et al., 2013.)

The Epoc device was used in a very effective, inexpensive way to control the wheel chair movement which produced very satisfactory results in the survey conducted by Stamps and Hamam (2010).

Mind Drone is one of the application of emotive device where it reads the brain signals from the user and control the motion of the drone. The drone acts as a cube in the emotive Epoc control panel application and using the thoughts of the users the drone can be moved in different direction. The Epoc device uses gyroscopes and facial expression measure to track the movement of the body and face expression and navigate the action like land or fly out the drone. There are several more independent researches going on using the Emotive Epoc device which are reported in the Emotiv official website. (Emotiv, 2016a.)

Having studied these above research paper closely about the Emotiv Epoc device it can be seen that it has been used for many different research purposes that involve the analysis of EEG signals. These studies also helped to understand the reliability and performance of the device in the field of software engineering. In the next section, we provide the methodology adopted in this thesis. The studies indicate the analysis is done using Emotiv tools like the Cognitive suite from Emotiv or Emokey or Emotiv SDK to perform the analysis. These are commercially paid tools which requires licenses to use the tools to perform the analysis. The Emotiv SDK provides a platform for developers to develop an API to conduct an analysis of the required emotions and cognitive measures (Klonovs & Petersen, 2012). To limit the scope of the study in this research we use the EEGLAB and MATLAB to perform the analysis. (Alshbatat et al., 2014; Emotiv, 2016a; Gomez-Gil, San-Jose-Gonzalez, Nicolas-Alonso, & Alonso-Garcia, 2011; Klonovs & Petersen, 2012.)
The other methods used for analysis using the Emotiv Epoc device is the machine learning algorithm like Support Vector Machine (SVM) to classify the data and identify the difference. This method requires the SVM algorithm to have a training set to analyze the signals and differentiate the signal for every participant. (Wang, Gwizdka, & Chaoratwitwongse, 2016; Fakhruzzaman, Riksakomara, & Suryotrisongko, 2015.)
3. Research methods

The aim of this thesis is to conduct a pilot study to understand the emotions and cognitive processes during software programming activities by employing EEG-based evaluation using Emotiv Epoc device and evaluation from self-reports derived from questionnaire data. The first research question aims to understand and apply the steps to be followed while using Emotiv Epoc device, starting from recording brain signal to analysis of the recorded data. For this, a literature review was conducted on the use of EEG and Emotiv Epoc device (see Chapter 2). In addition to that, in order to apply the methodology learned from the literature review, a pilot study was conducted. This pilot was set in such a way to help understanding the emotions and cognitive processes during software programming. Thus, two specific research questions were formulated, namely, RQ2: What is the cognitive and emotional activity as measured by different EEG-based and self-report measures of the participants when programming in two of programming languages of software engineering (C and Python)? and RQ3: What differences are observed between programming in C and Python with respect to programmer’s performance, emotions and cognitive load during programming?

For answering RQ2 and RQ3, the plan was designed, and conducted as an empirical study. In order to adopt an empirical study, the research should employ a particular study like surveys, experimentation or case study. In a quantitative based method, the data is expressed numerically (e.g. time in hours or line of code). To collect quantitative data, the employed method should involve measurement table, structured interviews, structured surveys or test instruments. A statistical tool should be used for the analysis of quantitative data. In a controlled experiment the advantage is that both control level and internal validity are high. This means that there is an advantage to establish a statistically causal relationship theory. The disadvantage of a controlled experiment is low representativeness, due to which results cannot be generalized. (Jedlitschka & Rombach, 2016; Rombach et al., 2007; Muijs, 2010.) However, in this experiment the aim is not to generalize the results from the finding. It only reports the finding for this pilot research.

The design of the experiment is a paired comparison with related sample set (counter balanced related samples), the participants are given with the same sample set. The participants perform the same task but with random assignment of order. The data is analyzed quantitatively (Jedlitschka & Rombach, 2016; Rombach et al., 2007; Muijs, 2010).

The experiment is designed to collect questionnaire data which is explained in section 3.1. To measure the cognitive and emotional level and statistically answer the research questions, EEG data is collected during the experiment. The brain signals will be converted into discrete format to analyze them numerically. In the section 3.2.1 the different emotion and cognitive measures that need to be analyzed are mentioned. Section 3.3 are the steps taken to plan the experiment with a pilot test in order to finalize the experiment steps. In the last section 3.4 are the steps to analyze the EEG data.

In order to maintain ethics in the experiment every participant is asked to sign a consent form before the participation. The interested participants were sent with consent form beforehand as seen in APPENDIX A. No harm was caused to the participants while recording the EEG signals and the data was analyzed anonymously.
The programming language chosen for the tasks are C and Python. The tasks are completed on an online IDE platform called Tutorialspoint\(^2\). To monitor and observe the participant’s activities during experiment a video camera was placed for recording. This only records the screen and usage of the tutorials provided for C and Python during the experiment.

### 3.1 Questionnaire design

In order to understand the participants before, during and after the experiment a questionnaire set is created to answer the research questions. The first questionnaire collects background information of the participant. This questionnaire is needed to assess participants’ experience, skills in programming and emotions about programming. The background information is collected prior to understand participants and their experience with programming in C and Python. A questionnaire for each programming language was created to understand participant’s emotions, difficulty level of task and experience with the interface. To understand the participants’ experience with Emotiv Epoc device placed on the scalp during the experiment, an overall experience questionnaire was designed. This was answered after completion all programming tasks as well as questionnaires for C and Python.

The design of the questions in the questionnaire related to emotions and cognitive measures are taken from the different research studies (Helle et al., 2011; Haapalainen et al., 2010; Berka et al., 2004). The questions asked are as described in the Appendix sections (C, D, E) and the results are presented in chapter 5.

1. Background Questionnaire (APPENDIX C)
2. Questionnaire Python (APPENDIX D)
3. Questionnaire C (APPENDIX D)
4. Overall Questionnaire (APPENDIX E)

### 3.2 Measures

In this pilot research, the data to be analyzed is based on the measures from EEG-signals and self-report. The section 3.2.1 provides the emotional and cognitive measures which are of interest in this study. The emotional states are measured by using the frontal asymmetry measure and cognitive dimensions like comprehensions, sustained attention and memory load are evaluated by using different frequency ranges. The self-report measures are explained further in the section 3.2.2.

#### 3.2.1 EEG based measures

Section 2.1.3 briefly explains various kinds of EEG based measures for the emotions and cognitive load. In this research, we are interested to find the emotional measures like motivation, happiness, frustration, stress, etc., which can be obtained from frontal EEG asymmetries.

\(^2\)http://www.tutorialspoint.com/
Alpha band power is used for calculating frontal asymmetry which is measured at the sites F3 and F4 as described in section 2.1.3. Alpha band frontal asymmetry is a measure typically used to capture emotional arousal or approach motivation (see e.g., Coan & Allen, 2004; Davidson, 1993; Rajanen et al., 2015).

In order to measure the cognitive load, the focus is capture the cognitive dimension like sustained attention, comprehension and memory load. Sustained attention and message processing are obtained by theta frequency band (4-8 Hz) which can be applied for the raw signals obtained at every channel location over the entire scalp area (Klimesch, Schimke & Schwaiger, 1994; Klimesch et al., 2001; Penolazzi, Angrilli & Job, 2009; Spironelli, Penolazzi & Angrilli, 2008).

The attention can be calculated using the alpha band (8-13 Hz) power over the entire scalp area (Doppelmayr, Klimesch, Stadler, Pöllhuber & Heine, 2002). Alpha band power can be calculated for specific scalp sites and frequency ranges to indicate specific measures of attention, thus frontal low-alpha band (8-10 Hz) power measured at F3, F4, and Fz are a measure for cognitive attention and message procession. The frontal upper-alpha band (10-13 Hz) measured at F3, F4, and Fz provides measures for cognitive processing related to recall performance (Klimesch et al., 1994; Klimesch et al., 2001; Spironelli et al., 2008; Rajanen, 2017).

In order to measure comprehension like deep understanding, the beta and gamma frequency band (4-50 Hz) is used at the frontal, parietal and central lobe (F3, Fz, F4, F8, C3, Cz, P3, Pz, P4, P8) (Lee et al., 2016). These measures of emotion and cognitive load could be used to answer the research questions 2 and 3.

The 14-electrode reference point in the Emotiv Epoc device as seen in section 2.2.1 does not provide electrodes placement sites at the central location like Fz, Cz and Pz. Hence while measuring the emotional and cognitive measures the other electrodes sites like F3, F4, F7, F8 regions are captured and analyzed. In order to obtain more accurate measures of the emotional and cognitive load other bio-medical EEG device like Advanced neuro technology (ANT) can be used.

### 3.2.2 Self-report measures

To answer the research question 2 and 3 we also use self-reports to measure and assess participants’ emotional and cognitive level. The data is collected in 4 different stages during the experiment. In order to assess the data descriptive statistics is used for analyzing the nominal, ordinal and ratio scale data in IBM SPSS (IBM SPSS, 2017). The frequencies in the descriptive analysis can be either absolute or relative and the measurement metrics like maximum, mean, count, etc., are used to represent the results sets in chapter 5. A parametric test is conducted using the Pearson correlation for comparison of two cognitive variables as seen in results section 5.2. (Jedlitschka & Rombach, 2016 and Rombach et al., 2007.)

**Background information**

Background information data aim to captures participants’ basic information such as age, gender, education level, nationality, skills of English language, highest educational degree, current enrollment degree at University of Oulu and about their programming exposure.
Ekman (1992) and Helle et al. (2011) explained different measures of emotions which have been used to capture the emotions using self-reports. The discrete emotions of positive and negative measures like happiness, relaxation, motivation, enthusiastic, boredom, frustration, idleness, stress and dissatisfaction. The various cognitive dimensions of measurement during programming are remembering syntax, comprehension of programming logic, etc. To understand the programming practices generally followed by the participants, we ask them questions about their motivation towards challenging problems, attentiveness during programming, interest and curiosity towards new programming concepts, usage of tutorials, etc., as indicated in APPENDIX C.

In order to measure the skills and work experience in programming, the questionnaire was designed to collect information regarding the skills in C and Python and their exposure to other programming languages. In order to measure their programming expertise, participants were asked about their software industry work experience.

At the end of the questionnaire an open-ended question was asked to understand the positive and negative aspects of programming tasks. All the above information captured from the above questionnaire will help to understand and mark a baseline to assess the background of all participants.

**Questionnaire in Programming**

In order to compare the participants’ usual programming behavior (captured from the baseline in the above questionnaire) and their behavior during this experiment, participants were assessed for the emotional and cognitive load immediately after the experiment task.

The cognitive measures like attention, comprehension and difficulty of tasks were assessed from the participant’s answers regarding easiness of task, their concentration level during the activity, time factor, etc. as indicated in APPENDIX D. The participants were asked to select the emotions that they encountered during programming.

**Overall experiment**

In the last questionnaire, the data collected are the cognitive measures about the overall experience of the programming tasks. A comparison is to be made in order to understand whether C is more difficult than Python, time is a factor in C and Python programming, comprehension measures like programming language syntax understanding, etc. as indicated in APPENDIX D.

In order to understand external threat to validity affecting the experiment, the information is collected from all participants regarding their possible behavioral changes caused by the use of Emotiv Epoc device.
3.3 Planning the Experiment

In order to test the steps and setup of the experiment, a pre-pilot was conducted where two members of teaching staff volunteered to participate. They had good knowledge of coding and had background knowledge in C and Python. The reviews from this test results were taken into consideration and the final steps were formulated as implemented in section 4.1.

Initial draft to collect the EEG signals are as mentioned in the below steps.

1. Setup the Emotiv device.
2. Explain the steps of actions to the participants.
3. Baseline recording: The participant is viewing at the blank screen/monitor for 3 minutes. This is to record the EEG signals when participant is relaxed.
4. 5 Coding tasks in C/Python for 25 minutes.
5. Questionnaire in C/Python
6. Baseline recording for 3 minutes
7. 5 Coding tasks in C/Python for 25 minutes
8. Questionnaire in C/Python
9. Overall Questionnaire

After observation and discussion with the test participants about the experiment, the problems encountered were noted. Both participants complained about the headset becoming uncomfortable after an extended period of time, namely after the second baseline recording, that is after about 30 min. This caused irritation and frustration for both participants and requested to quit just before the 5th task ended in the second programming session. The other observed factor was the device started to lose connectivity with the Testbench application causing loss of data. The problem could be that the electrode was dry. In consideration of these problems, new coding tasks were created; the first task was a combination of the original first two tasks. In the new task formulated, the participants were to complete statements like `scanf` for requesting name, enter two integer numbers and compare them. The total of statements were 5 in C and 4 in Python for the first task; the second tasks was replaced by the third task of test experiment which was to complete the program to print a pattern that represents a right angled triangle in number format until a series of 5. The total statements in task2 were 3 in C and 2 in Python; the fourth task of test experiment was replaced and by the third task and a minor change was made to the question. The new task was to enter 5 numbers in an array and swap the number in ascending order from low to high. In this task, a total of 5 statements in C and Python were to be completed. Though, some of the tasks were merged, the number of instructions to be completed per task was limited to maximum of 5 instructions per program. This also meant that the tasks were to be completed in 15 minutes for each programming language.

3.4 Data analysis

Here a brief flowchart showing the entire process of EEG data collection and analysis. Figure 4 represents the steps in this experiment to collect and analyze the data. The detailed description of all steps performed in pre-processing and analyses are explained in chapter 4 to the research question 1 about the overall process for data collection and analysis of the EEG data using Emotiv Epoc device.
Figure 4. EEG data collection and analysis.

The detailed implementation and explanation of the experiment tasks are described in chapter 4.
4. Implementation

In this study, the experiment is designed with various data collection methods. The different types of the data collected are from Emotiv Epoc device to understand the cognitive and emotional behavior, questionnaire data, video recording and the coding results of the participants. Section 4.1 explains the steps of the controlled experiment designed for the participants. The software and tools used in this experiment is further discussed in section 4.2. The self-report data is formulated into nominal, ordinal or interval scale type data to be analyze numerically using SPSS and the details are explained in section 4.3. In order to record the EEG data using Emotiv Epoc device the guidelines for recording are discussed in section 4.4. In the last section, the steps for the analysis of the EEG data using EEGLAB and MATLAB are formulated from various previous research.

4.1 Experimental Design

In this experiment, the programming session for every participant consists of a total of 6 tasks; 3 tasks in C programming and 3 tasks in Python programming. Using a random selection method three participants were to start programming in C and three participants in Python programming. Random ID were assigned to every participant to collect the data. The ID which contained P had to start the task with Python and participant ID with C started their task with C programming. Only one participant was allowed at a time in the lab to perform the experiment. The participants were given with suitable timeslot for attending the experiment. In between every experiment the minimum calibration window was set to 30 minutes to ensure the Emotiv Epoc device was charge and the guidelines were followed as instructed in section 4.4.

To ensure there is no interference with EEG data due to external factors as the experiment uses a wireless Bluetooth device. No participant was permitted to bring any electronic device into the lab.

During the experiment, no participants was given any access to internet for help with the tasks. The participants were given with a tutorial for both C and Python for syntax guide. The flow of experiment was explained in a Power Point Show where all links and instruction were provided to perform the tasks. The experiment had strict guidelines to complete all the tasks within 90 minutes and the experiment was planned with below steps.

Step 1: Consent form (APPENDIX A) signature: Every participant had to sign the consent form on arrival before starting the experiment. The consent form was explained for every participant before starting the experiment. On participation, they were given a free movie ticket.

Step 2: Set up of Emotiv Epoc device (15 mins): The Emotiv Epoc device needs to be placed on the participant. The time required for this was about 15 minutes. Before the participants arrived, the electrodes were made wet using the saline solution. The setup of device was performed as mentioned in the section 4.4. Once all the guidelines were followed, the experiment starts with step 3.

Step 3: 3 minutes’ baseline recording for watching blank wall: This is the first task of the experiment which is to collect baseline recording where the participant was instructed to look at the blank wall. The participant was asked to click on the power
point presentation to ensure they move to the next screen which is a blank screen. This click sound was to ensure it was the start of the baseline. Once the time was completed a click sound was made and the next slide with instruction to start C/Python program was given. The motive of this step is to ensure the participants are relaxed and the EEG signals are recorded to mark as baseline for the analysis.

Step 4: 15 min to perform coding tasks in Python/C: Start the programming tasks using the link given in the Power Point presentation. This redirects the participant to Tutorialspoint for performing the coding task mentioned in APPENDIX B.

Step 5: Answer the task-experience questionnaire: Once the coding task is completed in C/Python a link to questionnaire appears on PowerPoint show. This questionnaire consists of 6 questions. In this questionnaire, APPENDIX D every participant is asked to express the emotions and difficulty level of the programming task.

Step 6: 3 min baseline recording for watching blank wall: Repeat the baseline recording as mentioned in step 3. This is to record the benchmark collected for analysis.

Step 7: 15 min to perform coding tasks in Python/C: If the participants starts with C programming in step 3 then in step 7 the participant will start programming in Python. The link for Python tasks is provided in PowerPoint presentation which redirects to Tutorialspoint.

Step 8: Answer the task-experience questionnaire: Repeat step 5.

Step 9: Answer the overall experience questionnaire: This questionnaire consists of questions related to the overall experiment and related to the Emotive Epoc device. This is to collect the information about how participant felt while wearing the device and performing the programming task.

Step 10: Debriefing and feedback: In order to collect feedback from participants an open interview was conducted. In this task, focus was to collect information related to Emotiv Epoc device and if they felt any discomfort or harm when the device was placed for around 50 minutes.

The experiment was designed in a systematic format. The above steps were the guidelines given to every participant in a Power Point Show. This ensured that there was no interruption of any sort during the experiment. If the participants had any doubts during the experiment the participants were free to interrupt. To ensure all the parameters were checked and completed a checklist was created for the observer. The checklist (APPENDIX F) was to ensure the observer makes notes (tick mark) of timing of all activities in the experiment. The most crucial point in checklist was to ensure the marker points were set during the collection of brain wave signals. (Start Baseline, End Baseline, Start C task, End C task and same for Python). The other instruction for observer in checklist was to ensure the recording was started, consent form was signed by participants, time was noted for start and end of experiment and mainly the timer was set for the coding task of 15 minutes.

Camtasia application was started just before the start of experiment. The application has features to capture the screen and the object of focus. The camera placed only recorded the participant usage of the tutorial which was given during the coding task. This application also helps to track time which was helpful during data analysis. In order to view the participant another camera was setup during the experiment. This was to
observe and make notes about participants’ emotions states. Participants were instructed about the setup of the second camera placed for observation. It was also important to observe and report in checklist if participants are making movement during the baseline recording.

4.2 Materials and Equipment

The tasks used in the experiment are very important because it has a direct effect on the experiment results. The software used for programming tasks in this experiment is an online IDE tool called Tutorialspoint which supports various programming languages like C, C++, Java, Python, etc. to learn and practice programming (Free Online IDE and Terminal, 2017). The tasks for C and Python are designed using this IDE tool. The other software used are Microsoft Power Point Show to provide instruction to the participants, Emotiv Epoc for recording brainwave signals, Camtasia studio software for recording of screen and use of tutorial guide, Logitech camera for observational purpose during the experiment. The data seen from Logitech camera is used only for observational purpose and data is not recorded.

Tutorialspoint provide an online service to learn, test and practice programming. It has various features to save and share the project. Codingground3 in Tutorialspoint is the online IDE to practice tasks in C, Java, Python etc. Using the Codingground IDE the tasks were created for each participant to complete the program by filling the missing statements in program. Once the participants complete the coding tasks the data is stored in the Google drive. (Free Online IDE and Terminal, 2017.)

The questionnaire was designed using an online platform from Google services called Google Forms. The tool is very easy to design and prepare a list of questions required for the self-report. In total, every participant had to respond to four questionnaires. The data of the first questionnaire is collected prior to the experiment as mentioned in section 3.2. The responses from the participants are stored in the online service provider from Google called Google drive. The responded data is not editable and can be exported in an excel format. During the experiment, the participant responds to three questionnaires as described in section 4.1 step 5, step 8 and step 9.

### Table 2. Programming task difficulty level.

<table>
<thead>
<tr>
<th>Tasks in C and Python</th>
<th>Difficulty level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complete the following program to print your name (string), compare two integers values and then print if the variable is greater or smaller.</td>
<td>1</td>
</tr>
<tr>
<td>Complete the following program to make such a pattern like right angle triangle (see below) with a number which will repeat a number in a row for 10 natural numbers.</td>
<td>2</td>
</tr>
<tr>
<td>1 22 333 4444 55555</td>
<td></td>
</tr>
<tr>
<td>Write a program to sort the N natural numbers in an array and print in ascending order.</td>
<td>3</td>
</tr>
</tbody>
</table>

3 https://www.tutorialspoint.com/codingground.htm
The Table 2 provides the difficulty level of the tasks designed for the experiment. The difficulty was based on basic parameters of C and Python programming. In the difficulty level 1 the tasks were designed which had statement to be completed such as input output function. In difficulty level 2 the use of conditional statements to compares the values and also the use of single looping statement. The difficulty level 3 was more to understand if the participants concept for a scenario like using two different types looping and arrays to compare the values stored in different memory location. Overall the tasks were designed based on the learning pattern of concepts for a programming language. Since this study is about make an analysis of comparative study and understand their comparison on conceptual level we chose to the tasks chosen was same in C and Python. The randomness of tasks were made by allocating 3 participants in C and 3 in Python to start the programming.

The IDE interface where the participants performed the coding task is as displayed in Figure 5. Participants are requested to save the tasks after completing each task and then execute the task. If the three tasks or if fifteen minutes of allocated time is completed the participants are requested to stop and save the project. In order to share the code, participants were instructed to save the program and then click on the <Share code> as displayed in Figure 5. A sharable link is generated; This shareable link is to be provided in the questionnaire to identify the uniqueness of the participant. This data is further analyzed for correctness and completion of the task which is analyzed anonymously as mentioned in section 4.3.2.

![Figure 5. Tutorialspoint coding task online IDE.](image)

The programming tasks for C and Python which were given to the participants are listed in APPENDIX B. Each task is designed is such a way that participants are to complete the missing statements. The tasks were basic concepts of C and Python. Instruction for the program and hints were provided for every task of what was required to be completed in the program. The guidelines to complete each statement was given both in C and Python. The code marked in grey are the statement to be completed by every participant as listed in APPENDIX B. The task in C and Python were the same but only variance was the syntax and functionality of the program.
The online IDE tool has two prompts where user can enter data which are shell prompt and command prompt as indicated in Figure 5. The participants were asked to write the code in the command prompt. Next step is to save the program and follows the execution instruction given in the comments section to execute the program in shell prompt. Although there is an execute option in the interface it does not work well in the tool. These instructions were instructed to the participant before the start of the experiment. The coding task given to the participants did not have any syntax or semantic issue. The participants were only asked to complete the missing statement. The participants did not have any coding restriction like coding style for example in C program the looping statement used can be for, while or do-while loop or in Python to use the inbuilt function like sort function.

4.3 Questionnaire Data

The self-reports measures which are indicated in section 3.2.2 are analyzed using the SPSS tool. In order to analyze the data from self-report, the responded data needs to be assigned as nominal, ordinal or ratio. Nominal data is either the data which has numbers or symbols which belong to a particular class like country, name, sex, etc. Ordinal data is used when the data is organized in ranks like experience = {high, medium, low} or 3-point Likert scale or 5-point Likert scale. “Ratio is rank-order relationship with equal differences between the categories” (Armstrong, 2009) e.g. temperature scale in degree Kelvin. The data is first converted numerically as described in below sections for background, programming tasks and overall data responses and assigned to either nominal, ordinal or scale which is then analyzed using SPSS tool like descriptive statistics and Pearson correlation coefficient. (Jedlitschka & Rombach, 2016; Rombach et al., 2007; Stevens, 1946.)

4.3.1 Background information questionnaire

Due to the fact that there are no exact criteria for differentiating the skills of programming knowledge of the participants, we conducted a pre-survey based questionnaire. It was mandatory for the participant to fill in the information. The section 3.2.2 explains the importance for collecting the background information and the questions asked are in APPENDIX C.

The background information of all participants were collected and analyzed preserving the anonymity and confidentiality of the data. The responses were exported in an excel format and then data was pre-processed before analyzing it in SPSS tool. To analyze the data numerically the responses were to be represented in a numerical format. For example, the data for gender was pre-processed as “1=female” and “2=male”. The data was represented using a descriptive statistic using frequency method to represent factors like age, gender, skills in English language. The emotional states rated on a Likert scale is represented using bar graph and chart builder options as show in section 5.1.

4.3.2 Programming Task Questionnaire

To compare the emotions and cognitive level for programming it is important to design the questionnaire format as described in section 3.2.2. The random ID given to the participant was to be filled in the first question of the questionnaire. The link obtained from Codingground after the coding session is to be shared in the second question.
The cognitive measures were measured on a Likert scale with 1 as strongly agree and 5 as strongly disagree. The processing of coding tasks was performed by calculating the time spent on each task which is obtained from the video recording and the completion of the task statements from the IDE tool. A range was specified to measure the time spent on each task for example “0-3 mins = 1”, “3-6 min = 2”, “6-9 mins = 3”, “9-12 mins = 4” and “12-15 mins = 5”. The data was further processed using a descriptive statistical analysis as shown in section 5.2 for both C and Python programming.

In order to measure the participants’ performance, the video recording was utilized to identify the time spent on a task and the statements which were completed correctly. In this research scope, the correctness is not measured by the number of attempts made to reach a correct statement but only counts if the statements are correct or wrong. Since each task contains distinct set of statements; The total number of statements completed were identified for each task, e.g. Task1 in C contains 5 statements and assuming a participant completes 4 correctly a ratio is calculated by using the formula “number of complete correct statements/ total number of statements” which results to 4/5. This is performed individually for each participant and each task in C and Python. The overall task performance is calculated by adding the ratio of all three tasks and finding the average “C1+C2+C3/3”. A range is specified for the ratio obtained which provides the overall performance (Correctness and Completeness) in C and Python for e.g. “0-0.4= low”, “0.4-0.7=medium”,” 0.7-1=high” and the results are as indicated in section 5.2.

4.3.3 Overall Experience Questionnaire

In this questionnaire, the responses are collected after the participant completes all the programming tasks in experiment. As it was important to understand the effect of programming while wearing a headset device, participants were asked to rate their experience on 5-point Likert scale. To evaluate the cognitive measures the Pearson correlation coefficient between the attributes (Difficulty, Syntax understanding, Time) was measured using the ordinal values. As described in section 3.2.2, the questions were mandatory to complete. The responded data was analyzed numerical by assigning value as selected by participants and results are described in section 5.3. The data are either measured on an ordinal or ratio scale.

4.4 EEG recording

In the experiment, the Emotiv Epoc device was used to collect the brain signals. The collection of brain signals was done using the Emotiv Testbench method (Emotiv, 2016b). Using the Emotiv Epoc user manual setup guidelines, the below steps were followed to record and collect the data accurately. During the recording, the participants are asked to turn off all the electronic device as this could cause unwanted artifact to the signals as mentioned in section 2.1.3.

1. Make sure the headset is charged before use. The LED should indicate green when it is fully charged.
2. Hydrate the sensors with Saline hydration sensor pack. Add a few drops (6-7 drops if completely dry) on the sensor or hydrator pad, if the pads are not moist enough add few more drops to it.
3. Place the sensors units to the Epoc headset and make sure the clockwise lock is made.
4. Switch on the headset and pair the USB dongle with the headset for connecting the headset to the computer by inserting the USB dongle in the computer.
5. Place the headset on the study participant’s head. The black rubber sensor should be right below the ear lobe on both sides. The front sensor should be approximately three fingers above the eyebrows.
6. Check the signal quality and strength by using the Epoc Control Panel application. If all indicators are green then the connection with the scalp is successful, and good signal strength is expected. If an indicator is orange the connection is poor due to low hydration of sensors. If the indicators are red or black the detection is not possible for those electrodes.
7. If the electrode does not show any connection (red/black) repeat the steps to ensure the connection is green/orange. However, the aim is to have all signals green.

These steps are formulated from the user manual of Testbench which are mandatory to be followed (Emotiv, 2016b). Once the indicators are showing good connections, the device is ready for recording the signals. For the recording, Testbench application is used. On the start of EEG recording a marker window is selected. The marker window is used to indicate the activity as listed in Table 3. The marker setup helps to answer the research question 2 and 3 about what are the emotions and cognitive level during programming. This segregation of data would help in the analysis of the result.

**Table 3. Marker table used in Emotiv Testbench application.**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start baseline</td>
<td>1</td>
</tr>
<tr>
<td>Stop Baseline</td>
<td>2</td>
</tr>
<tr>
<td>Start P programming</td>
<td>3</td>
</tr>
<tr>
<td>Stop P programming</td>
<td>4</td>
</tr>
<tr>
<td>Start Baseline</td>
<td>5</td>
</tr>
<tr>
<td>Stop Baseline</td>
<td>6</td>
</tr>
<tr>
<td>Start C programming</td>
<td>7</td>
</tr>
<tr>
<td>Stop C Programming</td>
<td>8</td>
</tr>
</tbody>
</table>

The device operates at a standard sampling rate of 128 Hz (2048 Hz internal) and sampling method used is sequential. The frequency range for response is about .16-43 Hz. The minimum and maximum amplitude is set to default of 0 microvolt and channel spacing of 200 microvolts. High pass filter is selected to 200 microvolts. All channels are selected for recording. The data recording for the Emotiv Epoc device is stored in EDF format. (Nagy et al., 2014 and Emotiv, 2016b.)

4.5 Data Collection Problems

The experiment design was setup as per the guidelines mentioned in the section 4.4. In this experiment, we faced two kinds of technical issue which caused data loss. The intermittent connection problems occurred due to Emotiv Epoc device while recording the EEG signals and the participants not able to complete tasks due to crash of Online IDE tool. Below is the list of problems occurred.
1. Disconnection of all electrodes of the Emotiv Epoc device with Testbench application after the baseline recording for one participant.
2. One of the participants could not complete the task as the task failed to load during C programming. Hence it ended up that participant was assessed only on the two tasks for C programming.
3. Crash of the application; one participant used a wrong method in the code which due to which the looping statement did not have an exit and crashing the Codingground application.

The connection problems occurred with the EEG data collection were due the electrodes in touch with hair at few reference locations like FC6 and FC5. During the experiment two electrodes AF3 and AF4 were not connected as they were broken, hence making it impossible to collect the data at these sites as show in Figure 6.

![Figure 6. Electrodes having problem while collecting EEG signals using Emotiv Epoc.](image)

The individual challenges encountered while collecting EEG signals is listed below in the Table 4.

<table>
<thead>
<tr>
<th>Participant ID</th>
<th>Connectivity issues with Emotiv Epoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1P</td>
<td>AF3 and AF4 not connected as the electrode clamps were damaged.</td>
</tr>
<tr>
<td>2P</td>
<td>AF3 is not connected as the electrode clamp was damaged.</td>
</tr>
<tr>
<td>3C</td>
<td>AF3, AF4 were not connected as electrode clamps were damaged and partial loss of FC5 as device was not touching the head.</td>
</tr>
<tr>
<td>4P</td>
<td>AF3, AF4 were not connected as electrode clamps were damaged. In addition, all signals lost connection at 9th minute of the experiment. The reason could be that the device was not well hydrated. The participant head was small due to which the electrodes were colliding with each other in the frontal region. Previous study indicates with small head the data collecting was a problem (Wang et al., 2016)</td>
</tr>
<tr>
<td>5C</td>
<td>AF3 and P8 were not connected. AF3 was defective and P8 connection was terminated as there was no contact with scalp during experiment</td>
</tr>
<tr>
<td>6C</td>
<td>AF3 was not connected, FC5 and FC6 connection was lost during experiment</td>
</tr>
</tbody>
</table>

Due to these technical problems with the headset and connectivity, the data from the following 9 electrodes: F3, F4, F7, F8, T7, T8, P7, O1 and O2 were retained for analysis. Although data from 5 channels were lost, this will not affect the data analysis for analyzing the emotions and cognitive levels in this pilot experiment. The frontal asymmetry can still be calculated by finding the difference in following pairs (F3, F4 or F7, F8) (Reid, Duke, & Allen, 1998; Davidson et al, 1979). The cognitive measures like comprehension, sustained attention and memory load can still be measured as indicated.
in section 3.2. The analysis can still be conducted using the above healthy available electrodes signals.

### 4.6 EEG Data Analysis

EEG signals are recorded using the Testbech application and data is obtained in EDF file format. The EEG data obtained contains noise artifacts from the device and external disturbances like eye and facial muscle movement. As mentioned in section 2.1.3, artifact removal is a very important step to obtain the actual signals. The Emotiv Epoc device is a wireless BCI device which transmits data using a Bluetooth frequency. It is important not to use any electronic device during the experiment as it is possible that signals can be distorted by these electronic devices. To avoid the external artifacts the participants were instructed to switch off all electronic devices. The pre-processing of the EEG signals are very important to obtain the original signals to measure the actual emotions and cognitive levels.

In this research, MATLAB r2017a and EEGLAB\(^4\) version14 are used. The EEGLAB supports both EDF format and csv file format. The data can be converted from EDF to csv format using the Testbench tool. The comma separated values provide the numerical values of data.

The data recorded during the experiment are continuous EEG signals. To analyze the data, it is necessary that the data is divided into different epochs. There are four events in this experiment and four epoch are created with respect to these events. To identify these events the 8 references points were defined as markers to distinguish between baseline recording (two sessions marked by 4 markers at start and end of the baseline session) and programming sessions (4 markers to distinguish the start and end of each programming session). Thus, two epochs represent the two baselines and other two epochs represent the data while performing the programming tasks. The EEG data recorded while answering the questionnaires are deleted with this epoch creation.

The analysis for the different epochs is created based on the defined events in the experiment (baseline and programming). The baseline recording is recorded for 180 seconds and programming task for 900 seconds. Eliminating 5 seconds before and after to avoid the artifacts, the first baseline event epoch is defined from 5 to 175 seconds with reference to marker 1 and saved as epoc1. The programming event for Python is defined from 5 to 895 with reference to marker 3 and saved as epoc2. The same is done for second baseline with marker 5 and for the C programming with same interval with marker 7. This epoch creation is performed in step 11 for the data pre-processing.

The steps involved in preprocessing of the data using EEGLAB are (Ekanayake, 2014; Nagy et al., 2014; Trivedi, 2013; EEGLAB, 2017; Rodríguez, Rey, Clemente, Wrzesien, & Alcañiz, 2015.)

1. Open MATLAB and set the default path to EEGLAB.
2. Type <eeglab> in the command prompt to open the EEGLAB interface.
3. To import the data, select the import option and choose EDF format.
4. Select the participant’s EDF file for analysis.
5. Select the data and marker channels by specifying the range 1:20 in the Channel list box. The 20\(^{th}\) position includes the marker. The EEG recording also has other channel sets like GYROX and GYROY within the dataset (e.g. 18, 19

---

\(^4\) https://sccn.ucsd.edu/wiki/EEGLAB_TUTORIAL_OUTLINE
channel location). The best suited option is to remove the channel which are not useful in this analysis. The AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4 can be obtained by selecting a range from 3:16 from the EDF data.

6. Select only the data channels used by the Emotiv Epoc device by Edit > Select Data and adding in the channel range which are the 14 data channels: AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4.

7. Specify the Channel Location. For this step, use Edit > Channel location, and then Read locations from a pre-defined location (e.g. .locs file) file that defines the polar coordinates of each electrode position on the scalp as shown in Figure 7. This step is done to map the channel or electrode locations available in Emotiv headset on a 2D visualization in order to enable further the visualization of the heat maps of the EEG signals.

![Figure 7. 14 electrode channel location specified in EEGLAB.](image)

8. Next steps are for pre-processing of the raw EEG. First, filter the data using Finite Impulse Response (FIR) filter by passing the high pass filter of .16 Hz to remove the background signal. Another method is to use the IIR filter to track the background level and subtract the signals. These are two alternative steps which are mandatory to be done as suggested in Emotiv (2016b).

9. Next, one can apply a notch filter at 50 Hz to remove the interference caused by the power lines. To specify the notch filter, define the lower edge as 49 Hz and the higher edge as 51 Hz, and select Notch in the dialog window.

10. The filtering of the data is now complete, to view the data select plot > channel data(scroll) as shown in the Figure 8.

11. Next step is to define different Epochs based on the events defined in the experiment (baseline and programming tasks).

12. Repeat this step for each of the 4 Epochs.

13. The section 4.5 shows the problem which exists during the data collection and the healthy channel recording were chosen eliminating 5 channel recordings and analyzing 9 channels for all participants (F3, F4, F7, F8, T7, T8, P7, O1 and O2). The Figure 8 indicates the channels AF3 and AF4 had no data as it was a straight line for the entire dataset.

14. Next, Independent Component Analysis (ICA) is performed for the 9 channels. ICA is a very important step which is used to remove the artifacts like eye movements and muscle activities. Every individual has a different level of muscle movement and eye movement which causes artifacts. ICA is used to calculate for every individual these levels and individually eliminate the artifacts. (Delorme, Makeig, & Sejnowski, 2001.)
15. After running ICA, the data obtained can still contain artifact which cannot be removed by further filtering process and requires the signals to be analyzed by naked eye. The next step is to remove those artifact by eye which exceeds the 100 microvolts interval (Roach & Mathalon, 2008), as the data which are beyond +100 and -100 microvolts are not useful for analysis. The artifact removal process which are rejected by eye for the continuous data is shown in Figure 9. The artifact that can exist for the entire span of data set which needs to be selected by eye and rejected. Different studies employ different microvolt level thresholds e.g., +80 and - 80 (Mantini et al., 2007).

![Figure 8. Continuous EEG data after applying Notch Filter.](image)

16. The pre-processed data is now converted from continuous signals to discrete by using the time-frequency plot. This is performed on each channel to obtain the alpha, beta, theta and gamma frequencies. The following steps are performed in EEGLAB by selecting Plot > Time frequency transform > Channel time frequency.

17. The numerical values from the previous steps are obtained in a matrix format using a MATLAB script that calculates the power of the oscillations in each frequency band (alpha, beta, theta, and gamma) for each epoch.

All the above pre-processing steps from 1 to 17 are applied for each participant’s EEG data which are collected from Emotiv Epoc device. These data can be further analyzed using the SPSS tool to obtain measures of the emotional and cognitive processes under study (e.g., emotional arousal in terms of frontal asymmetry, comprehension, attention).
Other methods that can be adopted to analyze the EEG-data are machine learning algorithms like SVM, Open Vibe software, Emo-key, Emotiv SDK, which are used to find the differences in the emotional and cognitive measures as performed in numerous studies. (Alshbatat et al., 2014; Emotiv, 2016a; Gomez-Gil et al., 2011; Klonovs & Petersen, 2012; Fakhruzzaman et al., 2015.) The scope of the pilot study was limited to using the EEGLAB and MATLAB tools. In further studies this data can be used to analyze and find the psychological patterns of interest.
5. Results

The data collected from this experiment for analysis were self-reports, EEG, video captures, results of coding tasks and feedback from participants. In order to answer the research questions, the self-reports questionnaires data are analyzed using the IBM SPSS 24 tool. The data collected from Emotiv Epoc device were pre-processed for each participant individually using the EEGLAB and MATLAB but are not further analyzed, due to the fact that the data which is pre-processed may contain noise-artifacts which skew the statistically interpretation. Although the results in this thesis were not formulated for the EEG-data, it was very important to capture and understand its various measures which setup the pilot process for such further analysis. For example, one can employ ERP heat maps to understand the activity of cognitive states like memory load, comprehension and attention. However, due to the limitation of time and the scope of the pilot research, the EEG data were not reported using the ERP heat maps. The data captured in video recording is used to measure the time for which the participants spent on the coding tasks and identify the correctness of each task to measure the participant’s performance. To understand the external effects of the experiment, the data was collected from feedback session which focused on the factors like Emotiv device and the syntax tutorial provided. In this section, the formulation is made in accordance to the research method sections 3.2.2 and implementation section 4.3 of self-reports.

A background questionnaire was formulated in order to capture the overall experience of the participants in C and Python programming tasks. In the questionnaires, a Likert scale measurement was used for the range of selection: 1 = “Strong Agree” and 5 = “Strongly Disagree”. But during the analysis the Likert scale was inverted to represent the data in standard format, Accordingly, 1 = “Strongly Agree” as per questionnaire format was converted and interpreted as 5 = “Strongly Agree”, 2 =” Agree” was interpreted as 4 =” Agree” and so on. This was done for all the questions that involved a Likert scale measurement.

5.1 Participants

The data obtained from the questionnaire of background information provides a basis of participants’ skills and experience of programming. In this experiment, a total of 6 mentally fit students participated (4 males and 2 females). All the participants belonged to different nationalities: Finland, Bangladesh, India, Pakistan, Indonesia and Sri Lanka. They have all completed their Bachelor studies in their home country. Table 5 indicates the participants’ background information and percentage of overall participation. All participants are fluent or intermediate in English language skills. The motivation towards programming was either moderate or high. The factors like education background and nationalities are not considered in this study, as the scope of the study was to understand the emotional and cognitive states but it can be considered as a future study where the results can be generalized for a larger treatment of the sample set.

The skills in C and Python range from basic to intermediate as indicated in Figure 10. Among all participants one participant was enrolled in Computer Science master program and rest of the participants were enrolled in Information Processing Science degree program at University of Oulu, Finland. The experiment was conducted in the lab environment within University of Oulu campus. 50% of the participants learnt programming during the Bachelor studies and 16.7% at a secondary school level. Also, the answers reveal that 33% of participants learnt programming using online sources.
and books.

Table 5. Background characteristics of the participants, N=6.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender</td>
<td></td>
</tr>
<tr>
<td>Female</td>
<td>33.3%</td>
</tr>
<tr>
<td>Male</td>
<td>66.6%</td>
</tr>
<tr>
<td>Age Group</td>
<td></td>
</tr>
<tr>
<td>23 - 26</td>
<td>50%</td>
</tr>
<tr>
<td>27 - 30</td>
<td>33.3%</td>
</tr>
<tr>
<td>31 - 35</td>
<td>16.7%</td>
</tr>
<tr>
<td>English Skills</td>
<td></td>
</tr>
<tr>
<td>Fluent</td>
<td>66.7%</td>
</tr>
<tr>
<td>Intermediate</td>
<td>33.3%</td>
</tr>
<tr>
<td>Start Learning Programming</td>
<td></td>
</tr>
<tr>
<td>Secondary School</td>
<td>16.7%</td>
</tr>
<tr>
<td>Bachelors</td>
<td>50%</td>
</tr>
<tr>
<td>Self Learning</td>
<td>33.3%</td>
</tr>
<tr>
<td>Motivation</td>
<td></td>
</tr>
<tr>
<td>Moderate Motivation</td>
<td>50%</td>
</tr>
<tr>
<td>High Motivation</td>
<td>50%</td>
</tr>
</tbody>
</table>

Figure 10. Participants’ skills in C and Python.

The cofounding factor which could affect the controlled experiment is the participants’ experience in the software industry. 83% of the participants have worked for a software industry as shown in the Table 6 and Table 7 shows the number of years of experience in software industry. Although from the background data collected it does not show the experience factor in C or Python programming, in future studies this factor can be considered to understand the participants experience in the particular programming language to understand the effect of the experience on the coding tasks. The other factor which can be considered for the comparative study is the most recent hands-on experience of the programming language.
Table 6. Participants with work experience in software industry.

<table>
<thead>
<tr>
<th>Work experience</th>
<th>Frequency</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>5</td>
<td>83,3</td>
</tr>
<tr>
<td>No</td>
<td>1</td>
<td>33,3</td>
</tr>
</tbody>
</table>

Table 7. Years of experience in software industry.

<table>
<thead>
<tr>
<th>Years of experience</th>
<th>Frequency</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>1</td>
<td>16,7</td>
</tr>
<tr>
<td>Less than 1 year</td>
<td>1</td>
<td>16,7</td>
</tr>
<tr>
<td>1 - 2 years</td>
<td>3</td>
<td>50,0</td>
</tr>
<tr>
<td>3 - 5 years</td>
<td>1</td>
<td>16,7</td>
</tr>
</tbody>
</table>

The emotions expressed towards programming while participants code in a general approach is seen in Figure 11. The chart below indicates that all participants were highly focused, having a positive intent for programming with high motivation, enthusiastic, happiness and excitement. They have neutral feeling when it comes to negative emotions like frustration and stress. This chart indicated all the participants were passionate towards programming. Figure 11 is sorted in ascending order using IBM SPSS tool to indicate the ratings provided by participant on a Likert scale.

Figure 11. Emotions expressed by participates while coding in general. (5-point Likert scale: 1 Strongly disagree, 5 Strongly agree).
In order to understand the participants’ cognitive load in a general characteristic approach while programming, participants were asked to rate on a Likert scale about usage of tutorials while learning programming, focused on coding tasks with complete attention, etc. In Figure 12 the responses from the participants indicate the usage of the tutorials for learning are high and search for examples to improve the coding. The attention and focus of the participants are neutral towards programming and also, they lose the notion of time while programming at high level which provides a general idea of participants’ comprehension and memory load.

![Figure 12. Background of participant’s behavior towards programming. (5-point Likert scale: 1 Strongly disagree, 5 Strongly agree).](image)

### 5.2 Experimental Task Analysis

Once the programming tasks are completed the participant responds about the emotional states which he/she has encountered. This chart as shows in Figure 13 indicates the emotions felt by the participants during the programming task. The results show that the participants were highly frustrated in both C and Python task. In total, the number of participants in C programming task were more frustrated than in Python task. The participants were more focused in Python than in C programming. Excitement level was the same for both C and Python and motivation was more in Python than in C programming. The data interpreted from the chart indicates that Python tasks generated more positive emotions (Happy, excited, focused) than C tasks. None of the participants felt happy while coding in C programming language.
Figure 13. Participants expressed emotions using questionnaire after programming task.

The analysis made from the data collected when participants completed programming task to measure the cognitive levels are as indicated in Figure 14. The data was analyzed using the chart builder options in SPSS to plot a bar graph for parameters like ease of tasks, time constraints, etc. as shown in X-axis of the chat in Figure 14. The results show that C and Python tasks were equally interesting and rated just above neutral. The instruction given to the participants were very clear for both C and Python and highly rated which indicate the cognitive measure like comprehension. Time was a major concern as most participants did not find enough time for the given tasks which indicated the mental load dimension of the cognitive load. The easiness for C and Python tasks in the experimental setup was marked relatively low by the participants. It also gives a clear indication that the IDE used for the programming tasks was very easy to use, indicating that the IDE did not affect the performance of the participants. This measure expresses the cognitive measure of attention that participants did not concentrate on the IDE and were more into the tasks.
Figure 14. Comparison of C and Python. 5-point Likert scale: 1 Strongly disagree, 5 Strongly agree.

The descriptive analysis is made on the amount of time spent by the participant for each task as seen in Table 8. The time spent is measured on a scale from 0 to 15 minutes of total and an interval of 3 minutes as mentioned in section 4.3.2. On average participant completed more tasks in Python than C programming. In C programming the minimum time spent for task1 that is 6-9 minutes accounted for 50% of the participants whereas 66.7% of the participants spent the minimum time 3-6 minutes for Python task1.

Table 8. Time taken by participant per task, N=6.
This result shows that total time spent on C programming task1 was higher than Python programming task1. Similar assessment for task2 and task3 in C and Python revealed the time required and the frequency of completion as indicated in the Table 8 below. This also indicates that participants did not attempt few tasks in C as they had spent most of their time completing the first task.

Table 8 shows only the amount of time spent for each task, as the interest was not in the actual performance of the participants in programming and rather focus on understanding the emotion of the participants. Hence the total time provided for all the tasks was restricted to about 15 minutes. The data collected from the video recording and Codingground IDE was analyzed to understand correctness and completion of code written in C and Python. In APPENDIX B the statements highlighted in grey were to be completed participant in the experiment. The tasks focused on completeness of code rather than the preferred coding style of the participants. The Figure 15 shows the correctness and completeness of code for each participant without measurement of time as described in section 4.3.2. This chart indicates that participants performed significantly well in Python tasks in comparison with C tasks.

![Figure 15](image.png)

**Figure 15.** C and Python performance.

Comparing Figure 15 and Table 8, it is concluded that the participants could not only perform the Python tasks faster but also achieved higher correctness as compared to the C tasks.

### 5.3 Overall Experience Analysis

The correlation analysis was performed using the Pearson’s coefficient of correlation between C and Python for the attributes as seen in Table 9. The scale of the Pearson coefficient is from -1 to +1. If the correlation value is 0 then there is no linear association between the two variables. The positive (+) correlation indicates there exists a positive linear correlation and the negative correlation indicates there exists a negative linear correlation between the variables. The Pearson correlation coefficient is represented by the variable “r” during the analysis, p represent the significance test and
N is the sample size. (Sedgwick, 2012.) The Table 9 shows few attributes having a correlation and the attributes which are compared are overall difficulty level of programming language with respect to syntactical difficulty, can be interpreted as follows: There appears to be a high positive correlation between the two attributes \( r = 0.873, p < 0.05, N=6 \) proving that C is rated to be more difficult than Python with respect to the fact that the syntax of Python is easier than that of C.

There is a negative correlation \( r = -0.840, p < 0.05, N=6 \) between the two attributes - More time required in C than Python and prefer C over Python which proves the preference of C programming is strongly and negatively correlated to the time required for programming. Between the other attributes, there exists no significant difference as seen in the Table 9.

**Table 9:** Overall comparison of the programming task using two tailed comparison, N=6.

<table>
<thead>
<tr>
<th></th>
<th>C Difficult than Python</th>
<th>C is Easy to Understand than Python</th>
<th>More time required in C than Python</th>
<th>Syntax of Python easier than C</th>
<th>C and Python are Similar</th>
<th>Prefer C over Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Difficult than Python</td>
<td>Pearson Correlation</td>
<td>1</td>
<td>-0.234</td>
<td>0.603</td>
<td>.873</td>
<td>-0.245</td>
</tr>
<tr>
<td></td>
<td>Sig. (2-tailed)</td>
<td>0.656</td>
<td>0.205</td>
<td>0.023</td>
<td>0.640</td>
<td>0.457</td>
</tr>
<tr>
<td>C is Easy to Understand than Python</td>
<td>Pearson Correlation</td>
<td>-0.234</td>
<td>1</td>
<td>0.387</td>
<td>-0.166</td>
<td>0.070</td>
</tr>
<tr>
<td></td>
<td>Sig. (2-tailed)</td>
<td>0.656</td>
<td>0.448</td>
<td>0.753</td>
<td>0.895</td>
<td>0.390</td>
</tr>
<tr>
<td>More time required in C than Python</td>
<td>Pearson Correlation</td>
<td>0.603</td>
<td>0.387</td>
<td>1</td>
<td>0.322</td>
<td>-0.271</td>
</tr>
<tr>
<td></td>
<td>Sig. (2-tailed)</td>
<td>0.205</td>
<td>0.448</td>
<td>0.534</td>
<td>0.604</td>
<td>0.036</td>
</tr>
<tr>
<td>Syntax of Python easier than C</td>
<td>Pearson Correlation</td>
<td>.873</td>
<td>-0.166</td>
<td>0.322</td>
<td>1</td>
<td>-0.377</td>
</tr>
<tr>
<td></td>
<td>Sig. (2-tailed)</td>
<td>0.023</td>
<td>0.753</td>
<td>0.534</td>
<td>0.461</td>
<td>0.668</td>
</tr>
<tr>
<td>C and Python are Similar</td>
<td>Pearson Correlation</td>
<td>-0.245</td>
<td>0.070</td>
<td>-0.271</td>
<td>-0.377</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Sig. (2-tailed)</td>
<td>0.640</td>
<td>0.895</td>
<td>0.604</td>
<td>0.461</td>
<td>0.612</td>
</tr>
<tr>
<td>Prefer C over Python</td>
<td>Pearson Correlation</td>
<td>-0.380</td>
<td>-0.434</td>
<td>-0.840</td>
<td>-0.225</td>
<td>0.265</td>
</tr>
<tr>
<td></td>
<td>Sig. (2-tailed)</td>
<td>0.457</td>
<td>0.390</td>
<td>0.036</td>
<td>0.668</td>
<td>0.612</td>
</tr>
</tbody>
</table>

* Correlation is significant at the 0.05 level (2-tailed).

The rating provided on a Likert scale is seen in Figure 16 that participants strongly agree that more time was required to complete the tasks in both C than Python programming. In this pilot study, the participant’s rate that the syntax in C is more difficult than syntax in Python. The cognitive level of understanding is marked neutral which indicated that the understanding level in C and Python are same. The results also show that the participants prefer Python over C as the rating for the attribute C preference over Python is low. The below Table 9 also indicates in this study participant’s preference and favor was more positive towards Python than C programming. These results could vary based on the sample size and last usage of the programming language.
Figure 16. Overall comparison of programming task in bar graph 5-point Likert scale: 1 Strongly disagree, 5 Strongly agree.

Figure 17. Comparison of emotions from the background, C, and Python.

To analyze the emotions of participant’s emotional states before programming and after programming tasks. The data from the background questionnaire is compared with the data from emotional states obtained after programming in this experiment. In background information, the participants rated the emotional on a Likert scale from 1 to 5 (1 is strongly disagree and 5 is strong agree) whereas after the programming tasks the
participants were to select the emotions which he/she felt while programming. A direct comparison between the emotional states cannot be performed but Figure 17 provides valuable information about the frustration and motivation level of all participants during programming. The difference observed is participants were neutral about feeling bored and relaxed during programming but when they performed the tasks in this experiment shows they were not bored and relaxed in both C and Python tasks. The negative emotions like dissatisfaction, stress and idleness were very low felt during the programming tasks as compared to the general emotions while programming was neutral.

![Figure 18](image)

**Figure 18:** Emotiv Epoc device presence on participants, 5-point Likert scale: 1 Strongly disagree, 5 Strongly agree.

In Figure 18 the analysis was made from the data collected when participants completed their programming task. Overall the participants felt neutral about the presence of the device. Although all the participants felt the Emotiv Epoc device was disturbing, it did not effect their concentration while programming.

In the debriefing session, participants were asked with few open questions about attributes which could influence their programming performance as indicated in APPENDIX F. Since all participants were restricted to the usage of internet, participants were asked if the syntax guide provided during task were helpful. All the participants informed that syntax in Python was more helpful compared to C syntax guide and in Python they felt relatively easy to find information which was helpful for the experiment.

On an average, more number of participants indicated the Emotiv Epoc headset placed during the experiment was only felt during the baseline recording and not while programming tasks. Only one of the participants provided the feedback that usage of internet for the programming task could have been more helpful. Another feedback indicated that although writing the complete code was preferable than filling up code snippets, given the time duration of the experiment the tasks were easy to follow and complete.
6. Discussion

In previous research studies (Garcia et al., 2003; Lee et al., 2016) programming languages were compared using a quantitative study to understand the difference in programming style, runtime, correctness, line of codes, etc. or compared based on the expertise of the programmers in those programming languages. In this research, a comparison has been extended and performed based on self-reports and EEG data of programmers during coding in two different programming languages, namely, C and Python.

6.1 Answers to the Research Questions

In order to answer the research question RQ1 regarding the steps involved to record EEG data and analyze those signals, a literature survey was done to understand the usage and recording of EEG signals in Testbench manual (Emotiv, 2016b). In research study conducted by (Khushaba et al., 2013 and Esfahani & Sundararajan, 2012) the analysis was executed using EEGLAB and synthesis was performed using the machine learning algorithms developed to analyze the emotions. Using the above research as a basis, the analysis steps were formulated that particularly catered to the expected analysis method for our experiment, as listed in section 4.4.

The research question RQ2 aims to understand the emotions like motivation, happiness, focus, relaxation, stress, anxiety, dissatisfaction, etc. and cognitive dimensions like attention, comprehension and mental load of the programmers while coding. The experiment was conducted in a controlled environment with random assignment of same sample set of tasks in C and Python. With consent, the background information of participants was collected prior to the experiment. With the baseline set as “Basic” regarding skill level of participants in both programming languages, the results of self-report evaluation indicated the emotions and cognitive dimensions during the experiment.

Lee et al. (2016) and Müller & Fritz (2015) performed studies using different EEG devices in order to find the emotional and cognitive states observed during various activities in the field of software engineering. Lee et al. (2016) measures the cognitive levels like comprehension (beta and gamma frequency) during Java programming as seen in the related work section 2.1.4. Muller & Fritz (2015) measures the positive and negative emotions while performing change tasks as seen in the related work section 2.1.4. These studies list the important emotions and cognitive measures for which we obtain values during the course of the experiment in this thesis.

To answer the RQ2 using EEG data, data was recorded from Emotiv Epoc device using Testbench application. The data was further analyzed using EEGLAB and MATLAB to obtain the EEG signals of frequency ranging from 4-43Hz. However, the EEG data analysis for getting actual measure of emotions using frontal asymmetry (alpha band power) and sustained attention and comprehension (beta and gamma frequency) remains scoped as future work as it requires additional time to ensure the data are noise-free. Noise and artifact removal are crucial iterative steps in analysis of the EEG data (Daly et al., 2013 and Ranky & Adamovich, 2010) and evaluation of these processed signals after the noise removal would provide more accurate interpretation and results. The pattern of emotional and cognitive activation could have been observed further using the ERP heat maps, but were not reported due to the scope of the study being limited to statistical analysis of the data. The ERP heat maps would also require noise-free signals.
In order to answer the RQ2 using the self-reports, the questionnaire data was analyzed using IBM SPSS as mentioned in chapter 5, for the emotional and cognitive states expressed while programming in C and Python. The positive emotions expressed in C programming tasks like motivation, focus and excitement were at a near-average level on the scale used for measurement whereas negative emotions like frustration, dissatisfaction and stress ranked high. The cognitive dimension of comprehension indicates that the participants understood the tasks and were highly attentive, not bothered by factors like the programming IDE. Yet, due to the time constraint, the memory load was felt significantly high in the case of C programming.

The positive emotions expressed in Python programming were high focus, motivation and happiness whereas the negative emotions like dissatisfaction and stress ranged low. In Python, the cognitive dimension of comprehension indicates that the participants understood the tasks clearly and with high attention as the concentration was more on the programming tasks and not the interface. The memory load felt in Python was relatively less.

In the research question RQ3, the aim was to find whether any difference were observed between C and Python programming based on performance, emotions and cognitive load of the participants during the programming tasks.

- The first finding was that there was a significant difference along with a positive correlation observed in the cognitive load when the programming language syntax was compared with the difficulty level of the programming language itself using the Pearson’s correlation coefficient methodology.
- The second finding from emotional analysis indicated that although the participants were highly motivated, the restricted experiment environment and lack of help/resources during coding triggered frustration in the participants for both C and Python tasks.
- The third finding was that although the participants self-evaluated their programming skills in both C and Python as “Basic”, the participants performed significantly better in Python tasks than in C tasks.

The thesis thus captures important aspects of psychological activity in the daily lives of software engineers. From the results achieved by analyzing experimental data, we can correlate the effects of programming challenges in the emotional and cognitive behavior of the programmer and ultimately his daily life. The analysis in turn helps to understand the preferred choice of programming languages in software engineering owing to the emotional and cognitive comfort demonstrated by the programmers.

### 6.2 Implication of the Research

This pilot research is a particular comparative study of two programming languages commonly used in the field of software engineering. But, in general this finding can have a greater implication in software engineering where programming languages can be representative of not just its area of application but also its impact and acceptance within the software engineering community. An intuitive inference of the results achieved in this thesis reveal that structured programming language like C is not only more syntactically difficult than scripting language like Python, but also adversely affects the programmers’ psychophysiological behavior, especially in the case of novice programmers. This further implies that the trends in software engineering has seen a
shift towards scripting languages that are not only easier to grasp but may also be less emotionally challenging for the programmers.

Thus, this thesis sets the base for software engineering practitioners and researchers to develop more intuitive programming language concepts and constructs to eventually overcome adverse effects of emotional stress during coding. This would, at large, benefit the software engineering community as a whole and the software developers in particular.
7. Conclusion

In this thesis, a novel method was adopted to measure and quantitatively analyze EEG data in order to evaluate the emotions and cognitive levels of programmers while coding, using the Emotiv headset device and self-reports in an experimental setup. This Pilot research indicated that the Emotiv Epoc device is feasible to conduct a controlled experiment. Following the results of the experiment above, three distinct differences were observed in the cognitive and emotional measures of the programmers. Python programming languages was more preferred than C programming based on the syntax, difficulty level and time taken was also less in Python. The emotions in Python were more positive as compared to C programming. These variations eventually prove that if the EEG data analysis conducted in controlled environment can reveal exceptional conclusion about psychophysiological activities. However, in this experiment we do not aim to generalize the results and the evaluation are performed based on the pilot nature of the experiment. This particular thesis helped to formulate the steps that need to be followed in order to analyze the EEG data using the non-commercial tools like EEGLAB and MATLAB.

The challenges that were faced during the thesis work were due to the following limiting factors.

- Although, the data was successfully pre-processed with the EEGLAB to remove the artifacts, the data contained noise signals which could not be processed within the scope of this thesis.
- The next challenge that was faced during the experiment involved the usage of Emotiv device itself. As the device is very fragile, it needs to be handled with care. The device used in our experimental setup had two of its electrodes damaged and hence the signals from those respective nodes could not be captured. The other limitation to the Emotiv device is that it is a free size headset. This means that it can be either painful for broader heads or too big for electrode placement on smaller heads. If the electrodes are extra hydrated then there is possibility of having intermittent connection problem for the reference points, limiting the time constraints of the experiment.
- Due to the time limitation, the experimentation was conducted and restricted to six participants which could possibly cause a threat to external validity based on a smaller sample set.

The future scope of this research is to further analyze the statistical data for this research, the signals should be noise free. This requires further processing in order to convert the FFT signals to obtain numerical values for analysis. Once the numerical data is obtained for different channel frequencies (alpha, beta, theta, gamma) the average for baseline and experimental data can be calculated in order to compare the programming language. The data can be analyzed on more samples to generalize the result and avoid the threats to external validity for the controlled experiment in the related sample set. The experiment remains open to larger sample set which can help to avoid the external validity threat and ultimately produce generalized results.

This research works covers only one method that has been adopted to analyze the EEG signals obtained from Emotiv device. Further studies can be performed in order to compare and analyze the different methodologies adopted for such experimentation.
8. References


Jedlitschka, A and Rombach, D, (2016). The Empirical Method Course Study, Germany, TUKL.


APPENDIX A. Consent Form for Participants

Consent form for participating in a research experiment

Participant Number: ____________________________

In this Experiment, we will be performing a psychophysiological test which will measure your psychological and physiological activity while performing programming tasks on the computer. The measured physiological signal are brain signals. The psychological signals are measured using the sensors placed on scalp. The device used for measurement of signals is Emotiv Epoc Plus. During the experiment, you will also answer three questionnaires. Capture of screen will also be video recorded using the Camtasia software. The experiment will take in total about 1h30min. All data collected will be handled with preserving the anonymity and confidentiality. The results of data analysis will be reported in a master thesis. As incentive for participation in the experiment, you will get 1 movie ticket.

After the experiment has been completed you may want a more detailed explanation of why these measurements are made.

While we hope that you are going to perform the whole test, you have the right to suspend the test anytime.

I have read the above description of the experiment, and fully participate in the experiment voluntarily.

______________________________  ______________________________
Place and Date  Signature

______________________________
Name in block letters
APPENDIX B. C and Python Programming Tasks

C

/* Complete the program to request the user to enter name and two integer values for X and Y; */
* Print the name and compare the two values X and Y. *
* The message to be printed after comparison should be if X is greater than Y or X is smaller than Y; *
* To compile the code run the below commands in the shell prompt. *
* gcc Task1.c *
* ./a.out *
*
#include <stdio.h>

void main()
{
    char name[20];
    int x,y;

    // Fill in the missing statement to complete the program
    // Request the user to enter the name

    printf("Enter your name\n");
    scanf("%s", &name);
    printf("You name is %s\n", name);

    // Input two values to be tested
    // Use conditional statements to compare the values

    printf("\nInput an integer value for x: ");
    scanf("%d", &x);
    printf("\nInput an integer value for y: ");
    scanf("%d", &y);

    if(x > y)
    {
        printf("\n x is greater than y ");
    }
    else(x<y)
    {
        printf("\n x is smaller than y ");
    }
}

/* Complete the program, Save the file. Then execute the program. * Once the task is complete proceed to Task2.C */

/* Complete the Program to print a right angled triangle as below format with numbers. */
/* 1 */
/* 22 */
/* 333 */
/* 4444 */
/* 55555 */
* Use the concepts of function to complete. Call by reference method. *
* Looping concepts to print in triangle format. *
* To compile the code run the below commands in the shell prompt. *
* gcc Task2.c *
* ./a.out *
*/

Python

# The code should request the user to enter name and two integer values for X and Y; # It should print the name and compare the two values X and Y. # The message to be printed after comparison should be if X is greater than Y or X is smaller than Y; # To compile the code run the below command in shell prompt # << python Task1.py >>

# Request the user to enter the name
name = input(" Name of university is: ")
print("University is: ", name)

# Input two values to be tested # Conditional Statements to compare the two values.
x = input("Please type a number: ")
y = input("Please type a number: ")

if (x>y):
    print("X is greater than Y")
elif(x<y):
    print("x is smaller than y")

# Complete the Program, save the File. Then execute the program. # Once the task is complete, Proceed to Task2.py

# Complete the program to print a right angled triangle as below format with numbers. #
# 1 #
# 22 #
# 333 #
# 4444 #
# 55555 #
* Use the concepts of function to complete. Call by reference method. *
* Looping concepts to print in triangle format. *
* To compile the code run the below command in the shell prompt. *
* gcc Task2.c *
* ./a.out *
*/

# Complete the program to print a right angled triangle as below format with numbers
#
# 1
# 22
# 333
# 4444
# 55555
* Use the concepts of function to complete. Call by reference method. *
* Looping concepts to print in triangle format. *
* To compile the code run the below command in shell prompt *
* << python Task2.py >>
```c
#include<stdio.h>
// Fill in the missing lines to complete the program.
void main()
{
    int row=5;
    // Call the function with the number of rows to be printed.
    print(row);
}

print(int x)
{
    int n,i,n2=0;
    // Use the Looping statement to compute from the i values until x;
    for(i=1; i<=x; i++)
    {
        n2=n2+i;
        for(n=1; n<=i; n++)
        {
            // Print the appropriate variable which would give us the above right angle triangle pattern.
            printf("%d", n2);
        }
        printf("n");
    }
}
/* Complete the program, Save the file. Then execute the program
 * Once the task is complete, Proceed to Task3.C */

/* Complete the program to accept N numbers and arrange them in an ascending order
 * Use Looping concept to enter N number of elements
 * Sort the array using looping concept.
 * Below hints are provided which statements are missing
 * To compile the code run the below commands in the shell prompt.
 * gcc Task3.c
 * ./a.out
 */

#include <stdio.h>
// Fill in the statement to complete the program.
void main()
{
    int i, j, a, n, number[50];
    printf("Enter the value of N n");
    scanf("%d", &n);
    printf("Enter the numbers n");
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);
    // Use Looping statement to compare the first element of an array with second element of the array
    for(i=0; i<n; i++)
    {
        for (j = i + 1; j < n; ++j)
        {
            if (number[i] > number[j])
                { // Swap the numbers if greater using Temporary variable a.
        }
    }
}
# printTri is a function created by user. Define the printTri function in python.
# Complete the definition the <for loop statement> . Range is already specified.

Def printTri:
for i in range(1,n+1):
    print(str(i)*i)

printTri(5)
/* Complete the program, Save the File. Then execute the program.
 * Once the task is complete, Proceed to Task3.py */

#include <stdio.h>
// Fill in the missing lines to complete the program.
void main()
{
    int i, j, a, n, number[50];
    printf("Enter the value of N n");
    scanf("%d", &n);
    printf("Enter the numbers n");
    for (i = 0; i < n; ++i)
        scanf("%d", &number[i]);
    // Use Looping statement to compare the first element of an array with second element of the array
    for(i=0; i<n; i++)
    {
        for (j = i + 1; j < n; ++j)
        {
            if (number[i] > number[j])
                { // Swap the numbers if greater using Temporary variable a.
        }
    }
    // Use Looping statement to compare the first element of an array with second element of the array
    for(i=0; i<n; i++)
    {
        for (j = i + 1; j < n; ++j)
        {
            if (number[i] > number[j])
                { // Swap the numbers if greater using Temporary variable a.
        }
    }
}
```

The code above contains a C program that is intended to print a right-angled triangle pattern. It also includes a Python function `printTri` that prints a string repeated a certain number of times, starting with the number 1 and increasing by 1 each time, up to a specified number. The Python function is intended to be used within a C program to print the right-angled triangle pattern. Additionally, there is a C program that is intended to accept N numbers and arrange them in an ascending order using looping concepts. The Python code snippet is also intended to be used within a C program to sort the numbers entered by the user in ascending order.
```c
int i, j;
    number[i] = number[j];
    number[j] = a;
}
}
// print the number of an array after sorting.
printf("The numbers arranged in ascending order are given below \n");
for (i = 0; i < n; ++i)
    printf("%d", number[i]);
}
/* Complete the Program, Save the File. Then execute the program.
* Once the task is complete, Save the Project.
* Share the code using the <<Share Code>> and Paste the Link in the <<Questionnaire>>.
* Follow the Instruction provided in the power point presentation.
*/
```

# Complete the Program, Save the File. Then execute the program.
# Once the task is complete, Save the Project.
# Share the code using the <<Share Code>> and Paste the Link in the <<Questionnaire>>.
# Follow the Instruction provided in the power point presentation.
APPENDIX C. Background Questionnaire

Research on coding in different programming languages
Background information questionnaire

This questionnaire is a part of a research project at University of Oulu. The present questionnaire is to understand the participant background knowledge about programming languages. All the questions are mandatory to answer. All answers will be processed anonymously. Participant's research ID or name are needed to match the questionnaire data with the data collected during the lab experiment. Please answer the questions as truthfully and accurately as possible. Thank you very much!

*Required

1. Please enter your research participant ID number sent to you in the email or your full name. *
________________________________

2. What is your Gender? * Mark only one oval.

Male
Female

3. What is your Nationality (e.g. Finnish, Swedish, etc.) *
________________________________

4. Your Age range? (in years) * Mark only one oval.

Less than 22
23 - 26
27 - 30
31 - 35
Greater than 35

5. English Skills * Mark only one oval.

Native Speaker
Fluent
Intermediate
Very poor

6. What is your Highest Education Degree * Mark only one oval.

High school or Vocational School
Bachelor
Master
PHD
Other: ________________________________

7. Current Degree Program enrolled at University of Oulu * Mark only one oval.

Information Processing Science – Software engineering orientation (e.g., GS3D, European Master in SE)
Information Processing Science – Information systems orientation
Information Processing Science – no specific orientation yet
Computer Science degree program
Other: ________________________________
8. How did you start learning programming? (Select only one option reflecting either learning at school or self-learning.) * Mark only one oval.

- Primary School
- Secondary School
- College
- Bachelors
- Master's
- Self Learning (I have learned programming outside school)
- Other: __________________________

9. How motivated are you to learn a new programming language? * Mark only one oval.

- No Motivation
- Very Low Motivation
- Moderate Motivation
- High Motivation
- Very High Motivation

10. Please rate from your perspective the following statements about learning a programming language. * Mark only one oval per row.

<table>
<thead>
<tr>
<th>Item</th>
<th>Questions</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>It is too difficult to remember the syntax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>I do not understand the logic of the programming.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The materials for learning do not provide suitable examples.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>It requires certain skills that I do not have.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>It is not enough interesting, exciting, or provoking for me.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: In SPSS the 5-point Likert scale was coded 1: Strongly Disagree .. 5 Strongly Agree.

11. How do you generally feel while programming? * Mark only one oval per row.

<table>
<thead>
<tr>
<th>Item</th>
<th>Questions</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bored</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Happy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Frustrated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Relaxed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Enthusiastic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Idle</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Excited</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Dissatisfied</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Motivated</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Focused</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Stressful</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: In SPSS the 5-point Likert scale was coded 1: Strongly Disagree .. 5 Strongly Agree.
12. How do you generally approach a programming task? Please rate the following statements. * Mark only one oval per row.

<table>
<thead>
<tr>
<th>Item</th>
<th>Questions</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I usually use tutorials and examples.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>I focus on the coding with all my attention.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>I like to finish my coding task as soon as possible.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>I get distracted easily by other stimuli.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>I am usually interested and curious about the coding I am writing.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>I usually lose the notion of time when coding.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>I like to code challenging problems.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>I am not sure whether my code is producing the right result.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>I like to learn and try new things when coding.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>I finish the coding when I have given all my best.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: In SPSS the 5-point Likert scale was coded 1: Strongly Disagree .. 5 Strongly Agree.

13. How do you rate your programming skills in C ? * Mark only one oval.

- No skills
- Basic skills
- Intermediate skills
- Expert skills

14. How do you rate your programming skills in Python? * Mark only one oval.

- No skills
- Basic skills
- Intermediate skills
- Expert skills

15. Do you have experience with any other programming language * Mark only one oval.

- Yes
- No

16. Which of the following programming languages are you familiar with? * Tick all that apply.

- C++
- Java
- C#
- Linux
- Android
- Shell Scripting
- I am not familiar with any programming languages
- Other: ________________________________

17. Do you have any Work Experience with Programming in Industry? * Mark only one oval.

- Yes
- No
18. How many years of Working Experience do you have in Software Development in Industry? * Mark only one oval.

None
Less than 1 year.
1 - 2 years
3 - 5 years
Over 5 years

19. Are you interested in a programming job in industry? * Mark only one oval.

Yes
No

20. Please list the three most positive and the three most negative things about programming in general from your perspective. *

________________________________
________________________________
________________________________
APPENDIX D. Programming Questionnaire

Questionnaire in C/ Python.

The below questionnaire is for C programming. The same questions were used for python by replacing all instances of C to python.

This questionnaire collects data on your experience with the tasks in C. All the questions are mandatory to answer. All answers will be processed anonymously. Participant's research ID or name are needed to match the questionnaire data with the data collected during the lab experiment. Please answer the questions as truthfully and accurately as possible. Thank you very much!

*Required

1. Please enter your research participant ID number given in the email or your full name. *

2. Please paste the link below obtained after performing the C programming Task. *

3. Please rate the following aspects of your experience with coding in C. *

<table>
<thead>
<tr>
<th>Item</th>
<th>Questions</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The tasks were interesting.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>The tasks were too easy.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The programming interface was easy to use.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>I was concentrated on the tasks more than on the interface.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>The instructions were clear.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>There was enough time to complete the tasks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: In SPSS the 5-point Likert scale was coded 1: Strongly Disagree .. 5 Strongly Agree.

4. What were your feelings while programming the tasks in C? * Tick all that apply.

Happy
Relaxed
Excited
Frustrated
Bored
Stressful
Dissatisfied
Anxious
Motivated
Focused
Enthusiastic
Idle
Other: ____________________________________________

5. How would you rate your overall experience of coding in C? * Mark only one oval.

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negative experience</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Positive experience</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX E. Overall Experience Questionnaire

Research on coding in different programming languages Overall Experience questionnaire

This questionnaire collects data on your experience with the experiment. All the questions are mandatory to answer. All answers will be processed anonymously. Participant's research ID or name are needed to match the questionnaire data with the data collected during the lab experiment. Please answer the questions as truthfully and accurately as possible. Thank you very much!

*Required

1. Please enter your research participant ID number given in the email or your full name. *

__________________________

2. Do you usually compare programming languages while learning a new programming language? *Mark only one oval.

Yes
No

3. How would you compare C and Python? *Mark only one oval per row.

<table>
<thead>
<tr>
<th>Item</th>
<th>Questions</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C is more difficult overall than Python</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>It takes more time to write code in C than in Python.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>The syntax rules in Python are easier than in C.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Code in C is easier to understand than code in Python.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>I think C and Python are quite similar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>I prefer C over Python.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: In SPSS the 5-point Likert scale was coded 1: Strongly Disagree, 5 Strongly Agree.

4. Which programming language would you consider to work with in industry * Mark only one oval.

C
Python
C and Python
I am not interesting in programming at all
Any programming language
Other: ________________________________

5. Please rate your experience with the Emotiv Epoc headset. * Mark only one oval per row.

<table>
<thead>
<tr>
<th>Item</th>
<th>Questions</th>
<th>Strongly Agree</th>
<th>Agree</th>
<th>Neutral</th>
<th>Disagree</th>
<th>Strongly Disagree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I did not feel its presence.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>It changed my behavior during the experiment.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>It was disturbing while doing the tasks.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>I concentrated on the tasks that I did not feel any difference.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: In SPSS the 5-point Likert scale was coded 1: Strongly Disagree, 5 Strongly Agree.
## Checklist for Observer (Python first)

<table>
<thead>
<tr>
<th>Task of Activities</th>
<th>Check</th>
<th>Time</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>preps</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prepare electrodes before arrival of participant</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Consent form</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Presentation of timeline</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instruct the participants to leave their bag at the door and shut down all mobile and bluetooth devices</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instructing the user about the Power Point show</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instructing the user about IDE and <strong>share code button</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Setup of Emotiv Device</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Emotiv Setup(Signal Strength)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>participant arrives</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camera Switch ON ()</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>Start Recording(CAMERA)</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>Open Power point slide show and tell participant to wait until you say to proceed</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td><strong>Observer computer set up</strong></td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td><strong>Observer computer</strong>: Logitech Camera Switch On</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>Open marker file in test bench</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>Open timer website or mobile app</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td><strong>start experiment</strong></td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>Start EEG Signal Recording</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>Announce participant that he/she can start the PPS</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td><strong>Mark events during experiment and pay attention to participant</strong></td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>START BASELINE 1 Signal Sent(Emotiv)</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>STOP BASELINE 1 SIGNAL Sent(EMOTIV)</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>START <strong>Python</strong> CODING SIGNAL SENT(EMOTIV)</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>START TIMER AND NOTE TIME for Coding</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>STOP <strong>PYTHON</strong> CODING SIGNAL Sent(EMOTIV)</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>STOP TIMER AND NOTE TIME for Coding COMPLETE</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>START BASELINE <strong>Signal 2</strong> Sent(Emotiv)</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>STOP BASELINE <strong>SIGNAL 2</strong> Sent(EMOTIV)</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>START <strong>C CODING</strong> SIGNAL SENT(EMOTIV)</td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>START TIMER AND NOTE TIME for Coding in <strong>C</strong></td>
<td></td>
<td>!</td>
<td></td>
</tr>
<tr>
<td>Stop and save data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP CODING SIGNAL Sent(EMOTIV)</td>
<td>!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP TIMER AND NOTE TIME for Coding COMPLETE</td>
<td>!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop recording Emotiv</td>
<td>!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVE THE VIDEO FILE</td>
<td>!</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAVE THE CODING TASK Browser File in Drive</td>
<td>!</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Debriefing</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OPEN Interview Questions</td>
<td></td>
</tr>
</tbody>
</table>

**Open interview questions; write answers on the sheet**

- **DID you feel the Emotiv Device during the Exercise? When?**
  - START, MIDDLE, END OF SESSION? C VS Python
  - How did they affect?

- Cheat sheets usage, C, Python

- Any feedback from participant

- Which programming language was preferred and why