**sFree Apples: An Intelligent Tutoring System Integrating Kolb’s Experiential Learning Model as Pedagogical Module embedded with Facial Expression Recognition**

A Thesis

Presented to the

Department of Computer Science

Institute of Information and Computing Sciences

University of Santo Tomas

In Partial Fulfillment

Of the Requirements for the Degree

Bachelor of Science in Computer Science

By:

Borais, Alexavier B.

Francisco, Yves Rupert C.

Guazon, John Robert B.

Mabasa, Ryan Christopher G.

Adviser:

Seño, Jose L.

**December 2016**

**Approval Sheet**

**Thesis Title: Free Apples: An Intelligent Tutoring System Integrating Kolb’s Experiential Learning Model as Pedagogical Module embedded with Facial Expression Recognition**

**Proponents:**

1. Borais, Alexavier B.
2. Francisco, Yves Rupert C.
3. Guazon, John Robert B.
4. Mabasa, Ryan Christopher G.

In partial fulfillment of the requirements for the degree of ***Bachelor of Science in Computer Science***, the thesis mentioned above, has been adequately prepared and submitted by the proponents. This thesis was duly defended in an oral examination before a duly constituted tribunal with a grade of

**Ms. Charmaine Salvador-Ponay**

Panel Member

**Asst. Prof. Arne B. Barcelo**

Panel Member

**Mr. William A. Cortez Engr. Jose L. Seño**

Panel Member Adviser

Accepted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science.

**Engr. Jose L. Seño**

Chairperson

Department of Computer Science

Institute of Information and Computing Sciences

**University of Santo Tomas**

**Institute of Information and Computing Sciences**

**Department of Computer Science**

**Certificate of Authenticity and Originality**

We, the authors of the thesis entitled, “***Free Apples: An Intelligent Tutoring System Integrating Kolb’s Experiential Learning Model as Pedagogical Module embedded with Facial Expression Recognition***”, hereby certify and vouch that the contents of this research work is solely our own original work; that no part of this work has been copied nor taken without due permission or proper acknowledgment and citation of the respective authors; that we are upholding academic professionalism by integrating intellectual property rights laws in research and projects as requirements of our program.

If found and proven that there is an attempt or committed an infringement of copyright ownership, we are liable for any legal course of action sanctioned by the University and the Philippine laws.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Alexavier B. Borais**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**John Robert B. Guazon**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Yves Rupert C. Francisco**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Ryan Christopher Gulapa Mabasa**

Program: BS Computer Science Date: December 1, 2016

**ACKNOWLEDGEMENT**

The researchers would like to acknowledge the following individuals who contributed to the success of this thesis:

To our thesis adviser Engr. Jose L. Seño for graciously guiding and providing information throughout the research.

To our thesis coordinators Assoc. Prof. Perla P. Cosme and Ms. Ria A. Sagum for sharing their valuable knowledge and experiences.

To our panel members Ms.Charmaine S. Ponay, Asst. Prof Arne Barcelo and Mr. William A. Cortez for analyzing and imparting to us their ideas despite their limited time.

To our friends, thanks for all the support and understanding throughout those sleepless nights.

To our parents, we express our deepest gratitude for the love, understanding and patience that you have given to us.

To Almighty God, for his unending grace who made all of this possible. May this research bring glory to your name!

Table of Contents

[Chapter I 9](#_Toc468438152)

[The Problem and Its Background 9](#_Toc468438153)

[A. Introduction 9](#_Toc468438154)

[B. Background of the Study 10](#_Toc468438155)

[C. Theoretical Framework 12](#_Toc468438156)

[D. Conceptual Framework 17](#_Toc468438157)

[E. Statement of the Problem 20](#_Toc468438158)

[F. Objectives 20](#_Toc468438159)

[G. Scope and Limitations 21](#_Toc468438160)

[H. Significance of the Study 22](#_Toc468438161)

[I. Definition of Terms 23](#_Toc468438162)

[Chapter II 24](#_Toc468438163)

[Review of Related Literature 24](#_Toc468438164)

[A. Synthesis 36](#_Toc468438165)

[Chapter III 39](#_Toc468438166)

[Research Design and Methodology 39](#_Toc468438167)

[A. Hypothesis 39](#_Toc468438168)

[B. Research Methods 39](#_Toc468438169)

[C. Research Design 40](#_Toc468438170)

[I. Planning and Data Gathering 41](#_Toc468438171)

[II. Development and Evaluation 41](#_Toc468438172)

[III. Implementation 42](#_Toc468438173)

[D. Sampling and Data Gathering Procedure 42](#_Toc468438174)

[Chapter IV 44](#_Toc468438175)

[Presentation and Analysis of Data 44](#_Toc468438176)

[A. System Architecture 44](#_Toc468438177)

[B. Description of the Modules and Interfaces 48](#_Toc468438178)

[C. Test Results 53](#_Toc468438179)

[D. Analysis and Interpretation of the Results 68](#_Toc468438180)

[Chapter V 74](#_Toc468438181)

[Summary, Conclusions, and Recommendations 74](#_Toc468438182)

[A. Summary 74](#_Toc468438183)

[B. Conclusions 75](#_Toc468438184)

[C. Recommendations 75](#_Toc468438185)

[References 77](#_Toc468438186)

[Survey Questions 81](#_Toc468438187)

[User Interface 83](#_Toc468438188)

[Test Cases 97](#_Toc468438189)

[Exam Results 101](#_Toc468438190)

List of Figures

[Figure I - 1: Components of ITS (Tuaksubun & Mungsing, 2007) 13](#_Toc468439222)

[Figure I - 2: Architecture of the Distance Learning-Teaching Intelligent Tutoring System (Gascueña & Fernández-Caballero, 2005) 14](#_Toc468439223)

[Figure I - 3: Kolb’s Cycle of Experiential Learning (Kolb, 1984) 16](#_Toc468439224)

[Figure I - 4: Modified Architecture of the Distance Learning-Teaching ITS (Gascueña & Fernández-Caballero, 2005) 18](#_Toc468439225)

[Figure II - 1: Comparative Analysis of Pre-Test and Post-Test (Sampathkumar et al., 2014) 31](#_Toc468439250)

[Figure II - 2: FACS Example(Whitehill et al., 2008) 33](#_Toc468439251)

[Figure II - 3: - Affect-awareness Framework for Intelligent Tutoring Systems (Landowska, 2013) 35](#_Toc468439252)

[Figure III - 1: Rapid Application Development Process 40](#_Toc468439263)

[Figure III - 2: Exam Result 42](#_Toc468439264)

[Figure III - 3: Upper Limit 43](#_Toc468439265)

[Figure III - 4: Mean 43](#_Toc468439266)

[Figure IV - 1: System Architecture 44](#_Toc468439269)

[Figure IV - 2: Agent 46](#_Toc468439270)

[Figure IV - 3: Module 1 Exam Results (1-10) 53](#_Toc468439271)

[Figure IV - 4: Module 1 Exam Results (11-20) 54](#_Toc468439272)

[Figure IV - 5: Module 1 Exam Trend 54](#_Toc468439273)

[Figure IV - 6: Module 2 Exam Results (1-10) 55](#_Toc468439274)

[Figure IV - 7: Module 2 Exam Results (11-20) 55](#_Toc468439275)

[Figure IV - 8: Module 2 Exam Score Trend 56](#_Toc468439276)

[Figure IV - 9: Module 3 Exam Results (1-10) 56](#_Toc468439277)

[Figure IV - 10: Module 3 Exam Grade (11-20) 57](#_Toc468439278)

[Figure IV - 11: Module 3 Exam Score Trend 57](#_Toc468439279)

[Figure IV - 12: Module 4 Exam Grade (1-10) 58](#_Toc468439280)

[Figure IV - 13: Module 4 Exam Grade (11-20) 58](#_Toc468439281)

[Figure IV - 14: Module 4: Exam Score Trend 59](#_Toc468439282)

[Figure IV - 15: Module 5 Exam Grade (1-10) 59](#_Toc468439283)

[Figure IV - 16: Module 5 Exam Results (11-20) 60](#_Toc468439284)

[Figure IV - 17: Module 5 Exam Score Trend 60](#_Toc468439285)

[Figure IV - 18: Average Exam Grade (1-10) 61](#_Toc468439286)

[Figure IV - 19: Average Exam Grade (11-20) 61](#_Toc468439287)

[Figure IV - 20: Average Exam Grade For Each Module 62](#_Toc468439288)

[Figure IV - 21: Survey Question #1 63](#_Toc468439289)

[Figure IV - 22: Survey Question #2 63](#_Toc468439290)

[Figure IV - 23: Survey Question #3 64](#_Toc468439291)

[Figure IV - 24: Survey Question #4 64](#_Toc468439292)

[Figure IV - 25: Survey Question #5 65](#_Toc468439293)

[Figure IV - 26: Survey Question #6 65](#_Toc468439294)

[Figure IV - 27: Survey Question #7 66](#_Toc468439295)

[Figure IV - 28: Survey Question #8 66](#_Toc468439296)

[Figure IV - 29: Survey Question #9 67](#_Toc468439297)

[Figure IV - 30: Survey Question #10 67](#_Toc468439298)

[Figure IV - 31: Grades Interface 93](#_Toc468439299)

# Chapter I

# The Problem and Its Background

## Introduction

Programming is one of the most useful technical skills in the modern age. It leads to a great deal of opportunities in the digital world. As a result, learning to program can make an individual more productive, efficient and effective. There is vast amount of programming languages to learn, each having its own use, syntax, advantages, and disadvantages.

Learning a programming language can be intimidating as it may seem difficult, but with proper guidance and approach of learning, it can be enjoyable and convenient. This can be done through an innovative teaching-learning style.

One of the emerging approaches to teach students efficiently is the Intelligent Tutor System. According to the Valerie J. Shute and Joseph Psotka (1994), “*Our working definition of computer-tutor intelligence is that the system must behave intelligently, not actually be intelligent, like a human. More specifically, we believe that an intelligent system must be able to (a) accurately diagnose students' knowledge structure, skills, and/or styles using principles, rather than pre-programmed responses, to decide what to do next, and then (b) adapt instruction accordingly*.”

An abundance of research exists on the different implementations of the Intelligent Tutoring System on different subjects in the field of education. For instance, AutoTutor an Intelligent Tutoring System that assists college students in learning Newtonian physics, computer literacy and critical thinking skills by simulating the discourse patterns and pedagogical strategies of a human tutor.

Inspired from this idea, the authors of this study will use the Intelligent Tutoring System to teach students the Java programming language. An existing research authored by Lim-Rañola et. al (2015) which is “An Agent-Based Intelligent Tutoring System for Java Using the Experiential Gaming Model as the Pedagogical Module” is similar to this proposal, the difference between the aforementioned study and this study is the Pedagogical Module and the approach on enhancing the learning process of the student. Lim-Rañola et. al used Experiential Gaming Model as the Pedagogical Module and also integrated the concept of a game to motivate the students to learn.

The approach of the authors of this study is to use Experiential Learning as the Pedagogical Module. The agent will provide feedbacks in the exercise, and will be integrated with Facial Expression Recognition which will be used to assess the affective state of the student in the exam and serves as a basis on how the agent will provide hints.

## Background of the Study

An Intelligent Tutoring System is a computer system providing assistance, instruction or feedback to students in order to enhance the learning of the students, and at the same time, promotes interactivity between the student and the tutor.

The study focuses on the student and his needs. An approach to address this issue is the concept of *Experiential Learning*. According to Kolb (1984), Experiential Learning addresses the needs of the student, specifically the practice and hands-on experience which leads to an effective learning process.

According to Lewis and Williams (1994), Experiential Learning is quoted as “*In its simplest form, experiential learning means learning from experience or learning by doing*. Experiential education first immerses learners in an experience and then encourages and reflection about the experience to develop new skills, new attitudes, or new ways of thinking.”

Kolb’s Experiential Learning Theory involves a four step cycle. The learning phase may begin in any of the steps in the cycle depending on the condition of the student. One of the steps in Kolb’s Experiential Learning Theory is engaging on various experiences and activities. Followed by the observation and reflection part on which the student reflects on the experience that he had. Afterwards, the student contemplates and conceptualizes the theories and concept learned. Lastly, the student will test the new concepts he had learned on new experiences.

For a practical subject that is full of concepts, unless applied, will be difficult to understand especially for beginners and requires critical analysis from the student. The best approach would be Experiential Learning. Java programming or programming in general, is basically involves problem solving. Each individual learner will have unique and different approach in solving problems thus each learner will have different ways of perceiving the things they have learned from their experience. This promotes constructivism in which the student constructs knowledge out of their own experience. The researchers believe that Kolb’s Experiential Learning Theory, if not perfect, is the most suitable way of teaching Java programming.

In the modern world, the need for effective tutoring and training is mounting, especially in industry and engineering fields, which demand the learning of complex tasks and knowledge (El-Sheikh & Sticklen, 1998) Learning to program is a difficult task especially with no assistance. Comprehending concepts, understanding codes and commands and following different syntax requires practice. Self-learning a programming language is difficult. The authors of this study were inspired to create an Intelligent Tutorial System that will teach Java programming language to those who are willing to learn. This system will help build the foundation to learn other programming languages.

The study is an agent-based Intelligent Tutoring System that will use Experiential Learning as the Pedagogical Module to produce an interactive learning process and Facial Expression Recognition that will be used by the Agent to assess the affective state of the student (Kaufer, 2015). The objective of the study is to create an agent-based Intelligent Tutoring System that will help teach the Java programming language, providing practical experience and assessment of the affective state of the student which will lead to an effective learning process.

## Theoretical Framework

The theoretical framework of the study is based on the following theories: “Design of an Intelligent Tutoring System that Comprises Individual Learning and Collaborative Problem-Solving Modules” by Chitaya Tuaksubun and Surasak Mungsing and “Experiential Learning” by David Kolb.

According to Joseph et. al cited in the paper “Design of an Intelligent Tutoring System that Comprises Individual Learning and Collaborative Problem-Solving Modules” by Tuaksubun et. al (2007), there are five modules or components of the system where the expert model is an additional module to their system as shown in Figure I-1.

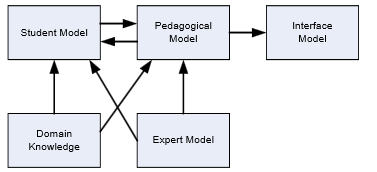


Figure I - 1: Components of ITS (Tuaksubun & Mungsing, 2007)

* **Student Model –** stores the information of the student as well as the feedback from the student.
* **Domain Model –** stores the content, lesson and teaching pattern.
* **Expert Model –** stores rule base in solving problems.
* **Pedagogical Agent –** responsible for retrieving the information from student model and expert model and process the appropriate content and activities.
* **Mentor Agent –** resembles the facilitator of ITS that is responsible for the transformation and storage of information as any parts of the system may require.
* **User Interface –** It is the bridge for communication between the student and the system and the system as well as reviewing the behaviors of the student. The information is sent back to the student module.

Figure I-2 illustrates the architecture of the distance-learning Intelligent Tutoring System and its main components.

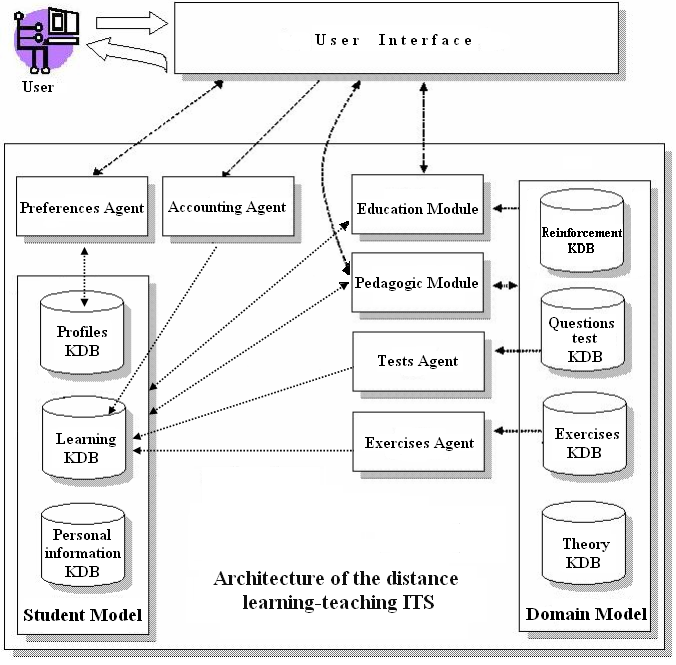


Figure I - 2: Architecture of the Distance Learning-Teaching Intelligent Tutoring System (Gascueña & Fernández-Caballero, 2005)

Firstly, the student reads lectures for the current topic, organized and presented by the Pedagogic Module through the interface. After reading the lectures, the student has to solve the proposed exercises prepared by the Exercises Agent. Afterwards, if the student finishes answering the exercise, the agent stores the exercise taken by the student in the learning KDB. After solving the exercises, the student has to solve the test questionnaire provided by the Tests Agent. Additionally, the Tests Agent stores the provided test questionnaire in the Learning KDB. If there are more topics to be discussed, the system goes back to presenting the lectures for the topics to be discussed. Otherwise, the student has finished studying the subject matter.

The three main modules in the Fig. 1-2 are the Student Model, Domain Model, and Pedagogical Model. The Student Model is composed of three knowledge database (KDB):

* **Personal Information KDB** – stores personal data of user to control access to the system.
* **Profiles KDB** – store the student’s level and learning style.
* **Learning KDB –** stores the exercise data such as tests and activities which have been taken so far by the learner and the metadata about the exercises i.e. the duration of the test etc.

The Domain Model is composed of four knowledge database (KDB):

* **Theory KDB –** incorporates the study material or pages of theory that have been prepared for teaching on subject matter.
* **Questions test KDB** - stores the set of test questions related to the subject matter.
* **Exercise KDB –** stores the set of exercises on the subject matter.
* **Reinforcement KDB –** comprises the data used by the Pedagogical Module to prepare the material to be presented when a student needs reinforcement.

The Pedagogical Model provides the functionalities to efficiently present the concepts to the student. This module or model has three main tasks:

* Provide learning guidelines and reinforcement or student.
* Update the statistics of the exercises and tests presented in the Domain Model.
* Store the reinforcement data into the Learning KDB and the responses or feedback given by the students.

Other components are Preferences Agent, Accounting Agent, Education Module, Test Agent, and Exercises Agent based from the architecture of the distance learning-teaching ITS (Gascueña & Fernández-Caballero, 2005).

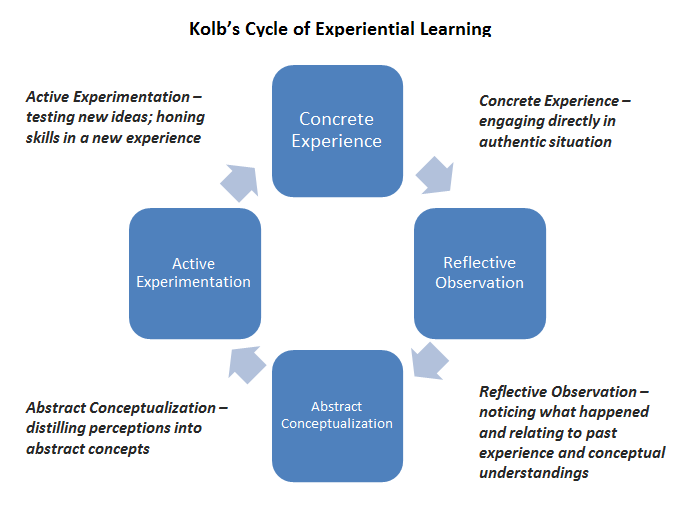


Figure I - 3: Kolb’s Cycle of Experiential Learning (Kolb, 1984)

Figure 1-3 illustrates the process of experiential learning. The four steps in the cycle are:

* **Concrete Experience (Do)** - a new experience of the situation is encountered, or a reinterpretation of existing experience.
* **Reflective Observation (Think) –**perceives the happening and relating it with the past.
* **Abstract Conceptualization (Conclude)** - reflection gives rise to a new idea, or a modification of an existing abstract concept.
* **Active Experimentation (Adapt/Test)-** the learner applies them to the world around them to see what results.

According to Kolb (1974), effective learning is witnessed when a person progresses through this cycle of four stages:

1. Having a concrete experience
2. Observation and reflection on that experience
3. Formation of abstract concepts (analysis) and generalizations (conclusions)
4. Which are then used to test the hypothesis in future situations, resulting in new experiences.

For this reason, effective learning only occurs when a learner is able to execute all four stages of the model. Hence, no one stage of the cycle is an effective as a learning procedure on its own. Learning is an integrated process with each stage being mutually supportive of each other. Kolb (1974) also adds that it is possible to enter the cycle at any stage and follow it through its logical sequence.

## Conceptual Framework

The study uses the Architecture of the Distance learning-Teaching ITS shown in Gascueña and Fernandez-Caballero and David A. Kolb’s Experiential Learning and to integrate the technology of facial recognition as shown in Figure 1-4.

conceptual-final.png

Figure I - 4: Modified Architecture of the Distance Learning-Teaching ITS (Gascueña & Fernández-Caballero, 2005)

The system incorporates Kolb’s Learning Theory as the basis of teaching student or as the pedagogy of the tutoring system since the learning model is actually a four-way process on how the students learn. Starting from Concrete Experience up to Active Experimentation, hence the cycle. Importantly, from its starting point, it must sequentially follow the four way process. The learning model is not the only factor affecting the learning process of a student. The affective state of a student can also be a factor in his learning process. The affective state of a person can be used to determine how a student tackles a concept. Through Facial Expression Recognition using third party software (Kaufer, 2015); the Agent will assess the affective state of the student in order to provide reinforcement to the student at a proper intervention.

Each concept will be divided into a module. A module will consist of a lesson, an exercise, and a single exam. Since the learning process can start at Concrete Experience (Do), each lesson has a brief basic introduction which will enable the student to read(Reflective Observation/Think), and formulate his or her idea(Abstract Conceptualization/Conclude)afterwards the student is given access to a video to enhance his knowledge while at the same time, and the student can also code to test his or her ideas (Active Experimentation/Test). This ensures that the student will learn through experience since he or she will be exposed to practical coding every exercise. If the student makes an incorrect code in the exercise, the agent will respond with a feedback as a form of tip or an error, and will not allow the student to proceed. Hence, the student is given the chance to experience and reflect on his or her answer, hereby completing Kolb’s experiential cycle.

In order to proceed to the next module, the students must complete all lessons and exercises, and also finish the exam with at least the passing score of for that particular module. Thereafter, the agent updates his or her progress. If the student fails the exam, the student may review his or her current module albeit with a different exam.

All students will start with the most basic module which will be the basic syntax until to the most difficult module which is objects and classes. A student cannot skip modules but can return to his or her completed modules for recap.

In addition, the agent will provide reinforcement, providing hints to which the student is struggling in the exam. The integration of the facial recognition serves as criteria for the agent to know when to provide reinforcement through hints. The facial expressions that will be analyzed by the system will be limited to neutral and anxious. The system will provide the option of hints to students whose facial expression is recognized as anxious. One attribute of an intelligent tutoring system is the capability to provide reinforcements to the student if needed. When a student encounters a wall or struggles to solve a problem, a hint can provide a recap of the student to solve that problem.

By implementing the Experiential Learning, the system provides an interactive learning process which will lead to an effective learning process

## Statement of the Problem

The study aims to develop an Intelligent Tutoring System for the Java programming language using the Experiential Learning Model and integrating Facial Recognition as factor in the adaptiveness of the system. The study aims to answer the following questions:

1. Will the facial recognition be an effective factor to determine when to provide hints to students?
2. Will Experiential Learning Model be an effective approach in teaching Java programming language?

## Objectives

The main objective of this study is to develop an Intelligent Tutoring System that will provide an effective and interactive learning process of the Java programming language. Specifically, the system should be able to perform the following:

1. The system should be able to appropriately provide a feedback which will assist the learning of the student in the exercise, and hints in the exam using facial expression recognition as a factor.
2. The system should be able to improve the knowledge of the student with regards to Java programming.

## Scope and Limitations

The study intends to build the basic foundation in Java programming of beginners. The target population consists of high school and college students.

Scope of teaching covers the following:

1. Basic Syntax
2. Data Types& Variables
3. Basic Operators
   * Arithmetic Operators
   * Relational Operators
   * Logical Operators
   * Assignment Operators
4. Conditionals
   * If statement
   * If-else statement
   * Nested if-else statement
   * Switch case statement
   * Relational expressions
   * Arithmetic expressions
5. Loops
   * While loop
   * For loop
   * Do-while loop

Limitations:

* Time constriction.
* The need for third-party software (video player, compiler, facial recognition).
* There is no limit to how many times a student can take up a module so if the student fails the same module multiple times, the student may lose motivation.
* The implementation of facial expression recognition.
* The student abuses the hints given by the ITS, losing the essence of learning.
* Extensive data gathering.

## Significance of the Study

The study will be of significance to the following groups:

1. ITS researches in the Computer Science community.

It will give a new perspective on the technology of ITS and doors to other future researches.

1. High school students/Non-technical Individuals

The implementation of the study will provide the students a way to build their basic foundation of Java programming.

1. College Community

The implementation of the study will provide an alternative method of teaching Java programming language to students.

1. Field of Computer Science

The implementation of the study will provide a new perspective on intelligent agents in the field of Artificial Intelligence.

## Definition of Terms

**Agent**- is a program or unit developed to react accordingly without the help of a human.

**Experiential Learning**– is the process of learning through experience, and is more specifically defined as "learning through reflection on doing".

**Facial Expression Recognition** –identify a specific individual in a digital image by analyzing and comparing patterns, and afterwards classifying their affective states.

**Intelligent Tutoring System (ITS) -** is a computer system that aims to provide immediate and customized instruction or feedback to learners, usually without intervention from a human teacher.

**Java** – a programming language developed and maintained by Oracle which is similar to C++ and C#.

**Knowledge Database (KDB) –** a database

**Pedagogy –**is a method or style used for teaching.

# Chapter II

# Review of Related Literature

This chapter contains the related literatures and ideas which serve as the basis for *Free Apples: An Intelligent Tutoring System Integrating Kolb’s Experiential Learning Model as Pedagogical Module embedded with Facial Expression Recognition* which contributes greatly to the study.

Experiential Learning is defined “as any learning that supports students in applying their knowledge and conceptual understanding to real-world problems or situations where the instructor directs and facilitates learning” (Wurdinger& Carlson, 2010). According to Kolb (1984), the depiction of the learning process is stated verbatim:

* Knowledge – “the concepts, facts, and information acquired through formal learning and past experience.”
* Activity – “the application of knowledge in a real world setting.”
* Reflection – “the analysis and synthesis of knowledge and activity to create new knowledge” (Indiana University, 2006)

Furthermore, Kolb’s Cycle of Experiential Learning involves: Active experimentation, Concrete Experience, Reflective Observation and Abstract Conceptualization. As stated in the University of Texas and Austin Learning Sciences (2015), experiences are chosen for their learning potential whether it provides for opportunities to practice and deepen emergent skills, encounter unpredictable situations that support new learning, or particularly learn from natural consequences, mistakes, and successes. As such, one of the facilitator’s roles is to choose appropriate problems, set boundaries, support learners, provide suitable resource, ensure physical and emotional safety, and facilitating the learning process.

Based from the work of Moon (2004), “*Experiential Learning can also be defined by what it is not, or how it differs from conventional academic instruction*.” In Experiential Learning, the context of learning is different since there are no academic texts or textbooks to study. Throughout the learning period, the student is in charge of his or her learning instead of being instructed in a classroom. As a result, the relationship between the student and instructor is difference, making it more student centered. Eventually, the student has to identify the knowledge they require and acquire it themselves, reflecting on their learning along the way.

With regards to the sample size, Hill (1998) stated “*determining sample size for an e-survey is not a cut-and-dried procedure.*” In other words, cut and dried can be understood as being clearly decided or settled in advance. In all cases of determining samples size, there is an element of arbitrary judgment and personal choice involved.

Furthermore in the article, Gay & Diehl (1992) states, *“in one way the typically smaller sample sizes used in applied or practical research have a redeeming feature*.” Their argument clearly states that large sample sizes enhance the probability of providing statistically significant results. With very large sample sizes, a significant result may yield a very small difference between the, and yet be of little practical use.

The study of El-Sheikh &Sticklen (1998) explains “The need for effective tutoring and training is rising, given the increasing complexity of the work place, and the knowledge-drain in contemporary commercial settings” particularly in the field of industry and engineering. Advancement in science and technology requires an individual capable of solving complex problems while operating and maintain sophisticated equipment. The advent of Artificial Intelligent and Computer Assisted Instructions gave rise to Intelligent Tutoring Systems (ITS) that can model learner’s understanding of the subject and adapt accordingly.

According to Lim-Rañola et al., in “An Agent Based Intelligent Tutoring System for Java Using the Experiential Gaming Model as the Pedagogical Module” (2015),the authors conducted a research on integrating the Experiential Gaming Model into an Intelligent Tutoring System. Scope of the teaching covered the following: Variables and Basic Commands, Operators and Expressions, Arrays, Conditionals, Loops, and Methods. Their method of teaching is simulated by a cookbook, the researchers applied a methodology of testing the students to define variables, create methods, use operators and expressions, and other instructions. The purpose of the gaming model is to keep the users motivated at the same time keeping the good quality of education.

In their first survey which composed of respondents that have taken up Java, 23 out of 25 respondents agreed to implement a game in the system. In their second survey, 20 respondents comprised of beginners and experienced Java users tested the said application with the game, 80% found the game to be motivating .As for the topics, their results revealed 68% of the respondents found Methods as the most difficult while Variables being the easiest with 16%. In conclusion, the authors found that the experiential gaming model has affected the interests and outcome of both beginners and experienced users positively.

As a recommendation, a dynamic learning environment was suggested in order to teach the Java programming language. It must be dynamic in a sense that it adapts to user’s need while at the same time providing reinforcement to the user.In addition, adding additional questions to the question pool, in order to decrease the chances of the student exhausting question pool. Also, the system can be improved by the method the system parses the users’ code to promote practice and error-detection less strict while providing an accurate feedback.

H. Chad Lane (2006) in his work “Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning”, the student does much of the work as possible while the tutor provides the appropriate feedback to lessen frustration and confusion (Merrill et. al., 1992). In addition, effective tutoring has less do with didactic explanations on the part of the tutor and more to do with the interaction between the tutor and student. In the work, the author remarked upon organizing the tutoring system:

*“One way to organize tutoring systems is around what role they are intended to play. At one end of the spectrum, some systems are intended to replace a textbook or classroom instruction to deliver domain content for the first time to a student (e.g., intelligent hypermedia systems). At the other end are systems designed to directly support practice (sometimes described as “homework helpers”). These usually complement an existing instructional component such as lectures. Although very few systems sit on the edge of this spectrum, ITS research tends to lean to the practice end. Indeed, practice is when “the rubber hits the road” in learning: it represents a volatile time when knowledge gaps are revealed and skills are automatized. Modern theories of learning stress the critical role of practice and most highlight the importance of feedback because of the risks of unguided learning (Kirschner et. al., 2006; Clark, 2004).”*

Moreover, the burden of creating an ITS was realize, hence approaches such as encoding expertise, teaching strategies, and domain models were proposed in the early 80’s order to reduce its development time. As such, authoring by example became a promising approach. The idea is that instead of encoding domain expertise and tutoring knowledge in an Artificial Intelligent programming language, the system demonstrates ideal solutions. The first one is to create feedback messages; the author specifies the message to the student at various points of the demonstration. Secondly, to handle mistakes, the author simply labels parts of the demonstration as errors then provides the appropriate feedback messages (H. Chad Lane, 2006).

Based on the study of Valerie Shute and Joseph Psotka (1994),an ITS behaves intelligently by assessing the student’s knowledge, the student model. Thereafter, the system considers what the student must know, the domain expert. Finally, the system decides the curriculum element (unit of instruction), the sequence of instructions and the means of presentation. From these characteristics, the system starts by selecting or generating a problem, subsequently works out or a solution through the domain expert or retrieves a prepared solution. The ITS compares its prepared solution to the student’s solution, and performs diagnosis based on the differences. Eventually, feedback is offered depending on the Student-Advisor decision – the last instance of feedback. After the feedback loop, the system updates the student skills model and increments learning process indicators.

As a supplement to the research paper of Valerie Shute and Joseph Psotka, the study of Albert and Schrepp (1999) conveys that an ITS is different from other forms of computer-assisted instruction due to its “intelligence”. First of all, the system can diagnose or assess the student’s knowledge base, utilizing its domain knowledge. Furthermore, it is adaptive to the student and provides an effective way of teaching the student. As a general consensus against the researchers, an ITS must at least contain four basic components: knowledge base, a student model, a teaching component, and diagnostic component.

* Knowledge Base - is also known as the expert system. This represents all the relevant knowledge of the domain of the particular subject.
* Student Model – represents the student’s cognitive abilities or skills, the overlay of the knowledge base. It shows the possible state of knowledge of the student in the course of his or her interaction with the ITS.
* Teaching Component – consists of the information from the knowledge base and student model and makes decisions regarding the appropriate strategies (instructions, exercises, or demonstrations). It contains a set of rules that determines each stage of the student model, specifically parts of the material that is relevant to the student.
* Diagnostic Component - A student’s state of knowledge is not recognizable since the hypothetical assumptions concerning the cognitive abilities or cognitive procedures are in the student model; this component infers the state’s interaction of the student with the ITS

The foundation of the research rests upon the general architecture of Gascueña and Fernandez-Caballero (2005). It is formed by the components: Student Model, Domain Model, and the Pedagogical Model. Additionally, there is an Educational Model to provide functions necessary for the teacher in utilizing the system. This model recommends how to enhance the structure of the presentation. Likewise, this module allows the teacher to change preferences, provide reinforcement, obtain statistical result, and consults the subject matter. Apart from the Educational Model, there is the Student Model that represents the body of knowledge towards the student whereas the Domain Model contains the knowledge about the contents to be taught. Lastly, there is the Pedagogical Model that incorporates four agents: Preference, Accounting, Exercises and Tests. It is the model responsible for the teaching strategy to efficiently present the subject matter.

Another example of an Affect-Sensitive intelligent system is the AutoTutor. D’Mello et al. (2008) proposed an enhanced version of AutoTutor that is Affect-Sensitive to the students by simulating a human tutor. An Affect-Sensitive system detects various emotions such as boredom, flow/engagement, confusion and frustration by taking into account conversational cues, body language, and facial features. The system was employed to assists students in Newtonian physics, computer literacy and critical thinking skills. The statement in the conclusion by the authors, “*The affect-sensitive AutoTutor aspires to keep students engaged, boost self-confidence, and presumably maximize learning by narrowing the communicative gap between the highly emotional human and the emotionally challenged computer.*” D’Mello et al. (2008) strongly points out that emotions are an important factor in ITS design.

“*One important lesson to learn is how students’ are sensitive to changes in the quality of the help provided.*“In developing an Interactive Learning Environment, the students are sensitive to the quality of help provided, and hints should be provided according to category that the student falls into (Arroyo et al., 2001).

In developing an ITS, the student’s learning style and collaborative learning must be considered. Hence, the study of Sampathkumar et al. (2014) considered the psychological aspects which influence the students’ learning behavior with Java as the course. The system employed the use of association rule mining and fuzzy C-means clustering. The system was composed of a student model, tutor model, and a system model. In their study, “The adaptation is derived into three levels of adaptation namely user level adaptation, user interface level adaptation and system level adaptation.” The user level adaptation would recommend variety of contents based on the individual learning behavior and overall performance of the student. As for the user interface level, it allows the student to personalize the preference for the user-interface. Lastly, the system level adaptation stores or replicates content in the distributed environment based on the current disk space and memory available. The results of the study revealed that the system can increase the learning factor of the student.

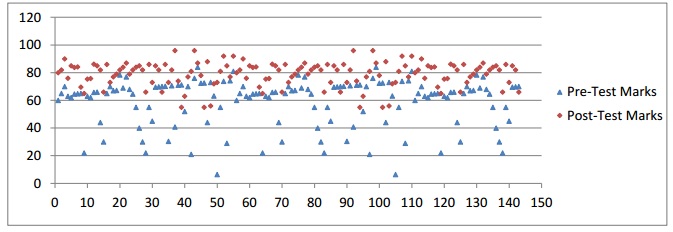


Figure II - 1: Comparative Analysis of Pre-Test and Post-Test (Sampathkumar et al., 2014)

ActiveMath is another instance of an ITS. It is a system that generates content depending on the specific pedagogical rules used and student data gathered. This includes changing the content presentation, the content itself, and interactive exercises according to the user’s preference and mastery level (Melis et al., 2001).

As stated in “ActiveMath: A Generic and Adaptive Web-Based Learning Environment” by Melis et al. (2001), ActiveMath is a generic web-based learning that adapts to the student by generating its content and changing the presentation to the desired or convenient preferences. The system gives specific course of materials depending on the goals, action history, student’s knowledge and pedagogical rules. The authors used a survey upon registration into their system to provide the knowledge level and the areas needed to be studied; afterwards, the course generator will generate a book which is a set of different contents arranged according to the initial survey. As time passes, data coming from the student model and profile will be used to generate interactive exercises. Similar systems include the use of static and dynamic fields in building the student model. In ActiveMath, static fields are used to hold the goal of the student. On the contrary, dynamic fields were used to hold the knowledge level, application level and action history, and to change them accordingly. The objective was to measure the progress of the student which is required for the adaptation of the ITS. More importantly, the separation of content and presentation is greatly needed. It allows the system to easily adapt on different preferences or settings. In addition, it generates content is helpful in speeding up the process of studying. Only the courses that are generated will be presented. Therefore, an increase of focus in specific areas will make the student master and fully understand the topic. Thus, it will be helpful in the development of future ITS especially when combined with emotion recognition to provide a more accurate and responsive system (Melis et al., 2001).

Tuaksubun and Mungsing (2007) in “Design of an Intelligent Tutoring System That Comprises Individual Learning and Collaborative Problem-Solving Modules” strongly points out that for the design of an ITS, there are two components that were considered. Firstly,an individual tutoring component provides appropriate lessons to the learner on the basis of his or her background, knowledge, interest and learning style prior to using the system. Afterwards, the agent monitors and processes these parameters in order to arrange the lessons for the learner. For the collaborative problem-based tutoring component, it provides tutorial problems and facilitates learners with hints for problem solving. Additionally, the system is designed to transition between modules during tutorial sessions at any time in order for the learner to review during brainstorming.

The study of Dewangan and Satao (2011) is a system similar to an ITS. In their study, the system employs different modules: learn phase, test phase, dictionary, and pronunciation section to teach the English Language. More importantly, the system has an intelligent feedback system which provides the user the awareness of his or her knowledge after an interaction with the system. This in turn, shows how valuable an intelligent feedback towards a student’s success in learning.

In the study of Whitehill et al. (2008), facial expression recognition is used to assess the following: difficulty of the lecture material in relation to the student, and the other is the preferred presentation of the materials. In their research, the authors experimented students using videos of people talking at different speed as the material. The goal was to analyze the facial expression to get the desired or preferred speed that is convenient for the student. The research used Facial Actions Coding System (FACS), a method used for facial expression recognition by classifying each movement of individual muscles in the face as action units as their method for the facial expression recognition.

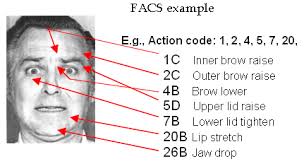


Figure II - 2: FACS Example(Whitehill et al., 2008)

Figure II - 2 shows the FACS method. The participants of this experiment were being analyzed real-time while doing their task. After the experiment, the results clearly show that it is possible to predict the difficulty of the material and the preferred speed of the video (in general presentation or settings of the material).

The research of Whitehill et al. group shows that facial expression recognition can greatly predict a person’s behavior and emotions, and can be applied to adapt or adjust to it accordingly.

Predominantly, an ITS focuses solely on the cognitive aspect rather than the emotional goal. As a result, the students are required to complete the task despite the fact that the students themselves are distracted. To state specifically, the systems were not adaptive in terms of emotional condition; hence, emotional states can support or suppress the learning processes, consequently strengthening the effectivity of the ITS. Also, it should adapt to the emotions of the students. In Landowska’s framework (2013) implemented in GERDA ITS that uses lexical analysis of user input, the Affect-Awareness framework provides intervention and reduces the risk of unnecessary interventions. It denotes that the program intervenes when needed.

There are three major components in GERDA ITS namely Affect-Recognition, Affect-Interpretation, and Affect-Aware Reaction, and is divided into subcomponents as shown in Figure II-3

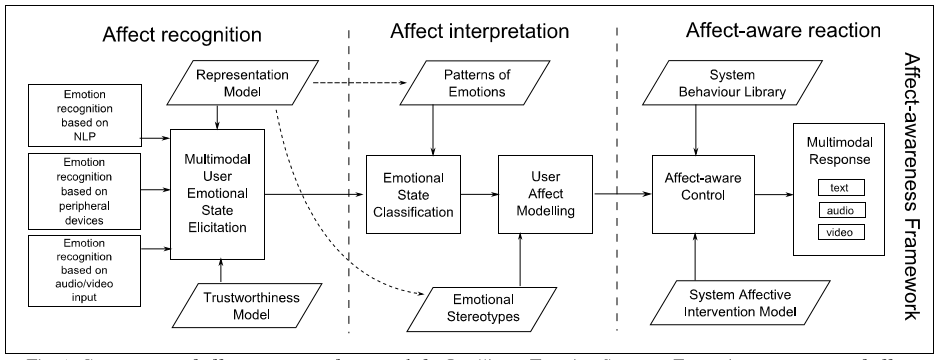


Figure II - 3: - Affect-awareness Framework for Intelligent Tutoring Systems (Landowska, 2013)

The framework accepts inputs from different devices. In the Affect Recognition, the Multimodal User Emotional State Elicitation handles the process for uncertainty and algorithm from the external models to provide an output that can be classified by the Emotional State Classification process. It uses the Trustworthiness Model for analyzing the inputs and finding its certainty value. On the contrary, the Representation Model provides a set of emotional states or values that can be classified later. Also, the models selected for representation can vary depending on the criteria.

For the Affect Interpretation, the emotional state is classified together with the patterns of emotions and the output from the elicitation. This yields to emotional states needed for making proper interventions and forwards it to the User Affect model. The model uses the Emotional stereotypes in order to identify the students or task stereotypes by clustering enough data of the students’ reaction and emotions.

Lastly, the Affect-Aware reaction is responsible for the response of the system. The System Affective Intervention model is used by the Affect-Aware control to reduce the risk of uncertain intervention. Moreover, the System Behavior library is used to map the evaluation to the specific reaction using a specific interface in Multimodal Response.

The results of the data implemented in GERDA proved that the framework helps in building an adaptive ITS that teaches students the same way as a teacher. However, the most important aspect of the framework is that it helps the ITS limits its interventions based on the certainty whether the hypothesis for the emotion of the student is highly true or not. As noted by Landowska (2013), it is not only limited to the ITS but also in other systems that uses Affect Awareness.

## Synthesis

|  |  |  |
| --- | --- | --- |
| **Title** | **Authors** | **Relation to the Study** |
| Experiential Learning Defined | University of Texas at Austin | The research explains Kolb’s Experiential Learning and the facilitator’s role |
| A Handbook of Reflective and Experiential Learning: Theory and Practice | Jennifer Moon | The research explains difference of Experiential Learning from learning in class and how the students acquire knowledge on their own. |
| What Sample Size is "Enough" in Internet Survey Research | Robin Hill | The research explains factors in determining sample size. |
| A Framework for Developing Intelligent Tutoring Systems incorporating reusability | Eman El-Sheikh and Jon Sticklen | The research provides a brief explanation of the need for an ITS |
| An Agent Based Intelligent Tutoring System for Java Using the Experiential Gaming Model as the Pedagogical Module | Evan Karlo Lim-Ranola,Wendell Nicole Tan, Nathaniel Francis Wicks | The researchers’ basis for the system. |
| Intelligent Tutoring Systems: Past, Present, and Future. | Valerie Shute, Joseph Psotka | The research shows how an ITS behaves intelligently. |
| Structure and Design of an Intelligent Tutorial System Based on Skill Assignments in Knowledge Spaces: Theories, Empirical Reseach and Applications | Dietrich Albert, Martin Schrepp | The research explains the basic components of an ITS and its functions. |
| An Agent-Based Intelligent Tutoring System for Enhancing E-Learning / E-Teaching | José M. Gascueña, Antonio Fernández-Caballero | The architecture serves as a basis for the researchers framework and components |
| AutoTutor Detects and Responds to Learners Affective and Cognitive States | Sidney D’Mello , Tanner Jackson , Scotty Craig , Brent Morgan , Patrick Chipman, Holly White , Natalie Person , Barry Kort , Rana el Kaliouby , Rosalind Picard, and Art Graesser | The research uses an Affect-Sensitive system with the use of facial expression recognition |
| Analyzing students’ response to help provision in an elementary mathematics Intelligent Tutoring System | Ivon Arroyo, Joseph Beck, Carole Beal, Rachel Wing, Beverly Woole | The research emphasized students sensitivity towards the quality of help received |
| Designing an Adaptive Distributed Tutoring System based on Students’ Learning Style and Collaborative Learning using Intelligent Agents | T. T.Sampathkumar,R. Gowri, Venkateswaran V. | The student’s learning style and collaborative learning must be considered. |
| Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning | H. Chad Lane | The research explains the role of the modules in the ITS and also its burdens. |
| ActiveMath: A Generic and Adaptive Web-Based Learning Environment | Erica Melis, Eric Andrès, JochenBüdenbender, Adrian Frischauf, George Goguadze, Paul Libbrecht, Martin Pollet, Carsten Ullrich | The pedagogical rules and formulas used in implementing the learning system combined with emotion recognition |
| Design of an Intelligent Tutoring System that Comprises Individual Learning and Collaborative Problem-Solving Modules | ChitayaTuaksubun, SurasakMungsing | The research explains that individual and collaborative learning must be considered. |
| Design and Implementation of English Language Tutorial System with Intelligent Feedback System | MeetaDewangan,K. J.Satao | The presence of an intelligent feedback system that helps students be aware of their knowledge. |
| Automatic Facial Expression Recognition for Intelligent Tutoring Systems | Jacob Whitehill, Marian Bartlett, Javier Movellan | Predicting difficulty level and preferred preferences of students by analyzing their facial expressions |
| Affect-awareness Framework for Intelligent Tutoring Systems | Landowska Agnieszka | A framework for detecting the emotional states of the students and make interventions. This framework is built with the goal of developing an ITS with emotional recognition. |

# Chapter III

# Research Design and Methodology

## A. Hypothesis

The system dynamically utilizes a lesson, an exercise, and a single exam. It is embedded with Facial Expression Recognition in order to effectively assist the student in his or her exam. Also, it will provide an interactive learning environment for students to learn Java programming language.

The researchers assumed the following given the necessary information for the study:

1. The Experiential Learning model will be an effective approach in tutoring.
2. The agent will be able to provide a feedback in the exercise and to provide reinforcement to the student in the exam through facial expression recognition.
3. The system will be able to improve the knowledge of students regarding the Java programming language.

## B. Research Methods

The researchers will use experimental method in order to monitor the effect of the system in teaching high school and college students. The experimental method is a controlled procedure; using a standardized approach, the researchers decide the location and time of the participants. As stated in the article by McLeod (2012), “It allows for precise control of extraneous and independent variables.” Thus, allowing a cause and effect relationship to be established. In this study, the independent variable is student while the dependent variable is the exam grade which differs depending on the independent variable.

## C. Research Design

The system is an Intelligent Tutoring System (ITS) based from Rañola et al., (2015) *An Agent-Based Intelligent Tutoring System for Java Using the Experiential Gamming Model as the Pedagogical Module* albeit the system serves an alternative method in teaching.

As a basis, the researchers gathered information regarding the architecture and basic components of an ITS. The system analyzes facial expressions during an exam evaluates the student’s performance at the end of each module which is the exam. But the most significant part is the design of the exercise in each lesson contained in the module which encourages Experiential Learning

It is built as a stand-alone application and requires a web server such as Apache in order for the facial expression recognition to function. The interaction between the student and the system occurs within the system such as the presentation of modules, viewing of grades, and etc. The system underwent these phases:

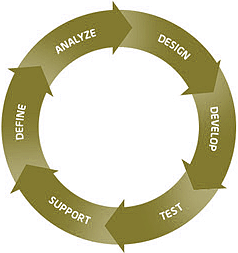


Figure III - 1: Rapid Application Development Process

Using Rapid Application Development, it decreased the development time, and at the same time, testing of errors (Appendix C) took place while building the system.

### Planning and Data Gathering

All information and brainstorming ideas are collected from:

* Consultation with the adviser.
* Past researches of developing an Intelligent Tutoring System, taking into the account the recommendations provided.

### Development and Evaluation

In development of the system, the following libraries/languages were used:

* Java –since it easier to grasp than C++ or C#, and is the most popular programming language. Therefore, it has many online materials that support the researchers.
* HTML& CSS – interface for the facial expression detection and presentation of modules.
* JavaScript – model and classifier for facial expression recognition
* PHP – bridges the facial recognizer to system.
* OpenHFT – a java runtime compiler
* SQLite - a portable database

With regards to the modules presented, mainly it was based from Oracle and Tutorialspoint and Java: A Complete Reference (Schilt, 2007) along with some information Malik’s book “*Java Programming: From Problem Analysis to Program Design”*.For assurance, the adviser was consulted in order to determine whether the lesson was appropriate. Moreover, a teacher evaluated the system whether it corresponds to Kolb’s Learning Model.

### Implementation

In this phase, the researchers will deploy the system provided that there are no more errors found in the test cases, and results provide that this is an effective system.

## D. Sampling and Data Gathering Procedure

Convenience sampling method is utilized in this study. The respondents are numbered at 20, must be in their first year college in order for them to qualify to answer. With regards to the students being sampled, the student will be interviewed to guarantee that he or she possesses no knowledge of Java, but expresses the intent of learning. Conversely, the student is required to interact with the system without taking into account age, gender, or other physical attributes. In determining the exam result, the following formula is shown in Figure III - 2:

Figure III - 2: Exam Result

As suggested from the adviser, the upper limit of the exam and the grading system will computed as shown in Figure III- 3:

Numberof Tries + 4)

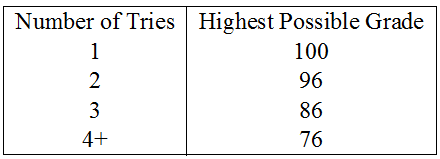


Figure III - 3: Upper Limit

For the overall percentage, this basic formula is used to compute the overall mean of the student:

Figure III - 4: Mean

# Chapter IV

# Presentation and Analysis of Data

## A. System Architecture

**Architecture-4-Final (1).png**

Figure IV - 1: System Architecture

The student creates a new profile on his or her first use of the program. Then, the student is prompted to input his or her credentials such as username, password, name, age, sex, school, and year level. When the student completes the registration, the Agent will insert all information to the User KDB which is in the student model.

After a successful registration, the student is given access to the system. The student provides his or her username and password in order to load his profile. The Agent checks if the credentials provided exist in the user database, and thereafter, it will retrieve the data of the student when found. Upon successful retrieval of data, the Agent will load all of the following:

* Student Grades from the User KDB
* Exercises from the Exercises KDB
* Modules, Text Lessons and Video Lessons from the Lesson KDB
* Practical Exercises, Exam Exercises, Challenges, from the Exercise KDB
* Exams from the Exam KDB.

The student is redirected to the home page once the loading is finished. Also, the student has an option to change the theme and feel of the system.

Initially, the student will only be able to access the first module and its contents. Upon opening the first module, the Agent will retrieve all the Text Lessons related to the module, eventually, it will show the first Text Lesson which is the introduction. Afterwards, the student can proceed to the next Text Lesson which is Lesson 1. Subsequently, the student is required to finish Practical Exercises for each lesson, in order for the student to experience hands-on coding in Java. Succeeding lessons will be unlocked by clearing the exercise for that lesson.

The exercises are composed of two parts, the first one is the video lesson and the second is the practical exercise itself. The student is given instructions on what he or she is supposed to do for the sake of clearing the exercise. The video lesson acts as a guide on how to code in Java and the practical exercise is the area wherein the student does the coding.

Checking-Practical-Print-Exercise-Final.png

Figure IV - 2: Agent

In the exercise interface, the student is given a text area, where he or she types the code. After submitting the exercise, the code can be checked by the compiler and the output can be viewed in the console. Finally, a response is given by the Agent if the exercise is correct or the Agent provides a tip or an error report whenever he or she makes a mistake in doing the assigned task. For an actual output, refer to Appendix B.

These work together in forming the dynamic tutoring system module in the system. If the student completes the assigned task, the exercise then locks and saves the code into the User KDB and unlocks the succeeding lesson in the module. This is done in order to give the student an option to review the past exercises completed.

After completing all lessons and exercises in the module, the Agent will unlock the exam for that module deeming that the student is already equipped with the knowledge to move on to the next module. The exam is made up of three sets. Each exam has a specific number of items composed of multiple choices about concepts tackled in the module and a coding exercise to go along with it. The time limit of the exam is 25 minutes. Each item is scored one point and five point for the coding exercise. The passing grade for the exam is 75%.

Before the start of the exam, the browser is opened by the Agent to initialize the Facial Expression Recognition (FER). This is used to capture the expression of the student in a span of time. As the student takes the exam, it writes the emotion in a text file and the Agent in return continuously reads this text file to determine whether the user is having difficulty in the exam. When the Agent detects the student’s emotion as anxious, it will respond by temporarily allowing the student to view the hint for that question. The hint only serves as a guide, but will never give the answer itself.

Once the exam is submitted or finished, the answers will be evaluated, graded and saved. If the student passed the exam, the next module will be unlocked. However, if the student failed, he or she would take the exam albeit a different set in order to proceed to the next module.

All of these are repeated until the all modules are completed. Furthermore, there is an opportunity for the student to take on a whole new module, the Challenges module. In this module, there will be a list of problems where the student has to solve each one of them by using known and existing algorithms in the programming world such as search algorithms, sorting algorithms, and etc. These problems are not graded and not required and is purely optional but serves as extra or additional knowledge for future purposes if ever the user decides to truly enter the world of programming.

## B. Description of the Modules and Interfaces

1. **Modules**

* Agent
* Retrieves all modules, lessons, exercises, and user credentials for the current user.
* Provides a feedback/tip for the user in the exercise.
* Updates the progress of the student in the Student Model
* Initializes the Facial Expression Recognition in the Exam
* Pedagogical Module
* Serves as the facilator in implementing the Experiential Learning. It presents all retrieved information from the agent as soon as the module starts.
* For the *observation* and *conceptualization* part, firstly, it will present an introduction to the module.
* After the introduction, the first lesson and the first exercise will be unlocked.
* In the exercise, there is a video lesson which serves as a guide in order to clear the exercise. In this part, the student gets to experience and reflect upon his or her own idea. If the student submits the wrong code/instruction, the agent will respond with a feedback/tip from the Exercises KDB in order to help the student.
* The student must finish all lessons and all exercises in order to take the exam. If the student passes an exam, he or she unlocks a new module. Therefore, the cycle is repeated until all modules are completed
* Student Model

User KDB

Exam\_Grades

User\_Exercises

id –unique identifier

userid – identifier of the user

exercise\_title – title of the exercise

code – the user’s code in the exercise

dateCreated – the time taken

dateModified – the time modified

Users

userid – unique identifier of the user

currentModuleId – the current module of the user

currentLessonId – the current lesson of the user

currentExamId – the current exam of the user

username – username of the user

password – password of the user

name – name of the user

age – age of the user

sex – male or female

school – current institution of the user

yearLevel – current year level of the user

Exam\_Grades

id – unique identifier of the exam grade

userid – identifier of the user who took the exam

exam\_title – title and set of the exam

raw\_grade – the raw score in the exam

totalItems – total number of items

percent\_grade – the transmuted raw score

status – identifies if the exam was passed or failed

dateTaken – the date of the exam

* Domain Model

Exercise KDB

Practical\_Exercise

lid – contains the unique id of the exercise

title –contains title of the exercise

tags –contains module and lesson number for categorizing

instructions – contains the instruction for the exercise

code – contains the starting code of the exercise

className– class name compiled by the compiler

methodName – method name compiled by the compiler

printValidator – check output in console

mustMatch – enablie case sensitivity

returnValidators – expected input and expected output

returnType – expected return data type

parametersType– data type parameters

explanation – response of the agent

Video\_Practical\_Exercise

lid – contains the unique id of the exam

title – contains the title and the set of the exam

tags – contains the module number for categorizing

videoLessonTitle – title of the video in the lesson

videoLessonId – identifier of the video in the lesson

practicalExerciseTitle – store exercise title (temporary storage)

practicalExerciseId – store exercise lid (temporary storage)

videoLessonUsing – retrieve title or lid

practicalExerciseUsing - retrieve title or lid

Exam KDB

Exam

lid – contains the unique id of the exam

title – contains the title and the set of the exam

tags – contains the module number for categorizing

exam – contains the questions to the exam

Lesson KDB

Text\_Lesson

lid – contains the unique id of the lesson

title – contains the title of the lesson

tags – contains the module and lesson number for categorizing

text – contains the path to the lesson

Video\_Lesson

lid – contains the unique id of the video

title – contains the title of the video

tags – contains the module and lesson number for categorizing

text – contains the path to the video

## C. Test Results

The researchers conducted a test on 20 samples composed of first year college students from BS Computer Science and BS Information Technology. The 20 samples were selected through Convenience Sampling. The samples used the ITS and went through five modules of basic Java. The following shows the results of each student.

Figure IV - 3: Module 1 Exam Results (1-10)

Figure IV - 4: Module 1 Exam Results (11-20)

Figure IV - 5: Module 1 Exam Trend

Figure IV-3 and 4 displays the results of the samples for each of their attempt in passing the exam of module 1 which is the Basic Syntax. Figure IV – 5 displays the trend of the average exam scores of samples that had taken the exam two or three times.

Figure IV - 6: Module 2 Exam Results (1-10)

Figure IV - 7: Module 2 Exam Results (11-20)

Figure IV - 8: Module 2 Exam Score Trend

Figure IV- 6 and 7 displays the results of the samples for each of their attempt of passing the exam of module 2 which is Data Types & Variables. Figure IV – 8 displays the trend of the average of scores of samples who took two times.

Figure IV - 9: Module 3 Exam Results (1-10)

Figure IV - 10: Module 3 Exam Grade (11-20)

Figure IV - 11: Module 3 Exam Score Trend

Figure IV-9 and 10 displays the results of the samples for each of their attempt of passing the exam of module 3 which is Basic Operators. Figure IV – 11 shows the trend of the average of the scores of the samples that had multiple attempt in passing the exam.

Figure IV - 12: Module 4 Exam Grade (1-10)

Figure IV - 13: Module 4 Exam Grade (11-20)

Figure IV - 14: Module 4: Exam Score Trend

Figure IV-12 and 13 displays the results of the samples for each of their attempt of passing exam of module 4 which is Conditionals. Figure IV – 14 shows the trend of the average exam scores of the samples that had multiple attempts.

Figure IV - 15: Module 5 Exam Grade (1-10)

Figure IV - 16: Module 5 Exam Results (11-20)

Figure IV - 17: Module 5 Exam Score Trend

Figure IV-15 and 16 displays the results of the samples for each of their attempt of passing the exam of module 4 which is Loops. Figure IV – 17 displays the trend of the average score of the samples who took the module 5 exam multiple times.

Figure IV - 18: Average Exam Grade (1-10)

Figure IV - 19: Average Exam Grade (11-20)

Figure IV - 20: Average Exam Grade For Each Module

Figure IV – 13 and 14 displays the mean of the grades of the samples in the exam for each module.

After they have finished with the system, the researchers asked the samples to answer a survey consisting of questions that will help the researchers in creating a conclusion and provide insights that will contribute to the study.

Figure IV - 21: Survey Question #1

Figure IV - 22: Survey Question #2

Figure IV - 23: Survey Question #3

Figure IV - 24: Survey Question #4

Figure IV - 25: Survey Question #5

Figure IV - 26: Survey Question #6

Figure IV - 27: Survey Question #7

Figure IV - 28: Survey Question #8

Figure IV - 29: Survey Question #9

Figure IV - 30: Survey Question #10

## D. Analysis and Interpretation of the Results

For Module 1, majority of the samples rated Module 1, Basic Syntax, easy to moderate in terms of difficulty of both exercises and exam as shown in the Figures IV-22 and 23. This is supported by the results in Figure IV-15 showing Module 1 having the second highest mean grade in exam. As observed in Figures IV-3 and 4, only one failed the exam two times before passing the exam, eight failed the exam once and eleven passed the exam in their first try 80 being the lowest score and 96 the highest.

As for Module 2, similarly to Module 1, majority of the samples found Module 2, Data Types & Variables, found it easy to moderate, slightly more moderate on the exam as seen in Figure IV-22. Though module 2 has the second lowest mean grade, Figures IV – 6 and 7 shows that only three failed the exam and only once.

Continuing on Module 3, majority of the samples found Module 3, Basic Variables, moderate in terms of the exercises and in terms of the exam most found it moderate – difficult as seen in Figure IV-22. In terms of average exam grade, Module 3 has the highest. In Figures IV – 9 and 10, it shows that one failed the exam thrice, one failed it twice eleven failed the exam once, and seven passed the exam in their first try in which all received a grade higher than 75 and from these seven three received a score of 100 resulting in a high exam grade mean.

On Module 4, majority of the samples found Module 4, Conditionals, difficult to very difficult, both in terms of exercises and exams. This can be seen in the average of exam grades of the samples seen in Figure IV-15 being the third lowest average grade which can be seen in Figure IV-22. Though it may seem that only a few have more than one attempt on the exam seen in Figures IV – 12 and 13. Majority of the sample received a grade ranging from 75 to 80 which is barely passing. Aside from that the lowest score that a student was able to get was 50. This supports how truly difficult the module is.

Lastly Module 5, Loops, was the most difficult module deemed by the samples as seen in Figure IV-22. This can be seen in Figure IV-15 having the lowest average grade. In Figures IV- 15 and 16, though only six failed the exam once and one failed the exam twice. It can be seen in the graphs that majority of the samples received less than 80 and only a four attained a grade greater than 80.

In each module there are samples who failed the exam once, twice, and thrice, in module 1 it can be observed that eight failed the exam once and one failed twice as discussed earlier. These are samples 1, 6, 9, 11, 12, 13, 16, 17 and 2 respectively. All the samples that failed once were able to pass the exam after taking it for the second time. Their scores in their attempts display an increasing trend, from 70 to 90, 73 to 86, 70 to 76, 53 to 80, and etc. Only one sample, sample 2, has a trend that showed to decrease at first then increased the next attempt. He failed the first exam with a grade of 66 then after his second attempt received a score of 63 then finally was able to pass with a grade of 86. Another good example that showcases a good learning trend is sample 3 in module 3, though he failed the module three times, Every time he reviewed the module, he was able to increase his score on his next attempt.

To better understand take a look at the Figures IV – 5, 8, 11, 14, 17, these figures shows the trend of the average scores of the samples who took the exams multiple times in Modules 1, 2, 3, 4, 5, respectively.

In Module 1, there are 9 samples that took the exam twice and 1 sample that took it thrice. Those who took it twice were able to increase their score from an average of 66 to 83. And the one who took the exam thrice had some up and downs, from 66 to 63 on his second attempt but was able to pass on his third attempt with a score of 86.

In Module 2, you can see that most of the samples that retook the exam only failed once. Their average score had a great improvement from 70 to 78.

In Module 3, you can see that there were quite a few who took the exam multiple times. 11 samples took the exam twice, 1 took the exam thrice and another 1 took it four times. Yet, you can see the gradual increase in their score in every attempt.

The same can be said be for modules 4 and 5, the average of the score of the samples had either a gradual increase or a dynamic increase in their scores.

From these interpretation of score trends for each module, the researchers can assume that Kolb’s Experiential Learning Model is an effective pedagogy that were able to help the samples improve for each attempt they make.

The survey questionnaires were used to gather insights from students that will contribute to our study. The first question is “Do you think the current setup is effective in teaching Java?” Note that the current setup is referring to the integration of Kolb’s Experiential Learning Model. Effectiveness can be a vague term but here we specified it on how effective the setup was in teaching the student or rather help them pass the exams. 20 out of the 20 samples responded that the current setup is effective for them.

Next is determining, “Which was the most effective part in the system?” Now there are two parts of the system on how the lessons were presented. First is the overview/introduction of the lesson and the other one is the video and practical coding exercise. The effectiveness was measured on which part they helped them understand the concept easier. 2 of the respondents answered that the overview/introduction was more effective than the video and practical coding exercise. The researchers may assume that these 2 that responded may be types of learners that prefer reading concepts rather than doing practical exercises. Another 2 of the sample responded that for them the video and practical coding exercises helped them understand the concepts better. These 2 that responded may be practical learners who would prefer doing exercise rather than read concepts. Lastly, majority of the sample which is 16 responded that both parts helped them understand the concepts better. Regardless of what type of learner they were or what they preferred, the combination of starting with presenting the overview/lesson then conducting an exercise with a video as a guide proved to be a helpful way of teaching topics.

A key component in the system was the Facial Expression Recognition (FER) in which it played a role in providing hints to students in exam. The samples were asked “Do you think that the hints provided by the FER in the exam were useful?” 13 of 20 samples responded that yes it was helpful to them but the remaining 7 responded otherwise. This shows that there is still a room for improvement for the accuracy of the FER. The 7 respondents that said the FER wasn’t helpful to them may not have been able to access the hints because of the inaccuracy or inconsistency of the FER. Though the FER was a third party software, it would be a good recommendation for future researchers to use an FER software with higher accuracy or if possible create their own FER software.

There are two ways on how the system delivers the lesson or concept, the overview/lesson and the practical exercise accompanied with a video as a guide. The samples rated both components. For the effectiveness of the overview/introduction of the lesson in teaching, six respondents rated it 10, seven rated it 9 and five rated it 8 and two respondents rate it 7 and 6. For the effectiveness of the practical exercise and the video guide, eleven respondents rated it 10, four rated it 9 and three rated it 8 and two respondents rate it 7 and 6. This results shows which part of the Kolb’s Experiential Learning Cycle had a greater impact on the internal cognitive process of the learner. The overview of lesson is associated with the Experience step in the Kolb’s cycle as for the exercise it can be regarded also as experience step or the testing step in the cycle depending on the learner.

User experience is a factor in the motivation and interest of student in using the system. As seen in Figure IV- 21, the researchers asked to rate the attractiveness and user friendliness that could have contributed to their UX and affecting their motivation. Seven samples rated it 10 out of 10, six rated it 9, 3 rated it 8 and 7, and only one rated it with a 5. Fairly, the users found the system to be simple and easy to understand on what to do even without the instructions of the researchers. Also their UX is a big factor in their motivation when using the system. No user would be motivated to learn if the system is confusing and disorganized.

The researchers asked the samples to rate each module on how interesting they are to them as seen in Figure IV - 24. Overall, the module 2, Data Types and Variable, seems to be the most interesting for most students. Next, is module 4, Conditionals, module 5, Loops, module 3, Basic Operators and lastly Module 1, Basic Syntax where a few found it not interesting at all. Their interest on a particular module has an effect on their performance on the exam for the module. As you can see in Figure IV-3 and 4 that a few of the samples had failed the exam once or twice, this can be related to their interest on the module Basic Syntax. For the most interesting module according to the samples which is module 2 Data Types and Variables, you can see the effect of this in Figure IV- 5 and 6, you can observe that most of the samples passed the exam in their first try and only three failed once.

One way to determine that the Experiential Learning was successful or effective is that if the learners are interested and motivated in their learning. This can be seen in Figure IV – 25 that the majority of the samples find Java very interesting after learning it. This can be interpreted that the samples are willing to learn more advanced Java programming.

The factors that we have considered to determine if the system is effective is the exam results of the students, each module has a few sample that failed once or twice yet they still are able to pass it. You can see the upward trend of their results and only one sample had a learning trend that had a downward trend but still was able to pass and increase his score. Those who failed their exam simply had to repeat the learning cycle by reviewing the module which resulted to a higher score in their second or third attempts.

Other than learning trend, we considered the satisfaction of the users since one of the ways to measure how effective a system is to let users experience it and be the judge. And lastly as discussed earlier, one of ways to determine that an effective or a successful experiential learning occurred is that the system is able to garner the interest of the learner.

# Chapter V

# Summary, Conclusions, and Recommendations

## A. Summary

*Free Apples* is an Intelligent Tutoring System developed by the researchers to teach the basics of Java programming to those willing to learn. Free Apples utilized Kolb’s Experiential Learning Theory as the pedagogy or approach of teaching.

The researchers decided to integrate Kolb’s Experiential Learning Cycle within the system to simulate the four-way process while teaching Java. Each topic is separated into modules. Each module contains lessons, exercises and an exam. Practical exercises are accommodated with a video tutorial; on the other hand, the agent provides a feedback whether the submitted code is correct or incorrect.

At the end of each module, the student will be required to take an exam which will unlock the next module. If the student fails, he or she can retake the exam albeit a different set. During the exam, the student will be assisted by the Agent within the system. The Agent will provide hints to the student depending on the facial expression.

           Kolb’s Experiential Learning Cycle is integrated in the learning process of the student being immersed in various exercises they gain new experience. From these experiences, the students reflect on their experiences. From their reflection, they conceptualize new theories. Furthermore, the students are given the opportunity to apply these new theories in their next exercises, exam or in the challenges presented to the students.

## B. Conclusions

As shown in the results, the system has been proven effective in teaching Java in terms of how the teaching material was presented and also proven effective in terms of interest after learning it. All of the students interviewed and selected have no prior knowledge in Java, but expressed intent in learning. They were immersed with the course provided by the ITS, and still found Java to be interesting after learning it

The researchers were able to achieve their primary objectives: to improve the knowledge of students with regards to Java, to provide a feedback from the Agent, and to provide reinforcement from the agent if needed using Facial Expression Recognition. Although the students found the exercises and exams difficult, the researchers concluded that Experiential Learning is an effective approach in teaching Java.

## C. Recommendations

For future developers of an Intelligent Tutoring System:

* Add more sets of exam to avoid students exhausting all the exams and retaking them. This will also provide students more opportunities.
* Add visual aid in presenting lectures, there are some student that prefer to watch someone explain to them concepts rather than read it themselves.
* Provide a visual agent to increase motivation.
* Provide collaboration exercises to also build up soft skills of individuals such as teamwork and communication.
* Add audio in video tutorials to provide better instructions.
* Add more feedbacks provided by the agents.
* Improve accuracy of feedbacks of agents.
* Improve accuracy of reinforcement/hints by the agent provided it is not abused.
* Add the functionality of tracking the strength and weakness of a student to the agent.

# References

Albert D., and Schrepp M., (1999). Structure and Design of an Intelligent Tutorial System Based on Skill Assignments in Knowledge Spaces: Theories, Empirical Research and Applications. *Knowledge Spaces: Theories, Empirical Research and Applications,* 179-196. Retrieved fromhttp://css-kti.tugraz.at/research/cssarchive/publicdocs/publications/file1260877867.pdf

Arroyo et al., (2001). *Analyzing students’ response to help provision in an elementary mathematics Intelligent Tutoring System*. Retrieved from http://users.sussex.ac.uk/~bend/aied2001/arroyo.pdf

Dewangan M., and Satao K., (2011). Design and Implementation of English Language Tutorial System with Intelligent Feedback System.*International Journal of Scientific & Engineering Research, 2(8),*Retrieved from http://www.ijser.org/researchpaper%5CDesign-and-Implementation-of-English-Language-Tutorial-System-with-Intelligent-Feedback-System.pdf

D’mello et al., (2008). *AutoTutor Detects and Responds to Learners Affective and Cognitive States*. Retrieved from http://affect.media.mit.edu/pdfs/08.dmello-etal-autotutor.pdf

El-Sheikh, E., and Sticklen, J. (1998). A Framework for Developing Intelligent Tutoring Systems Incorporating Reusability. IEA-98-AIE: 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Benicassim, Castellon, Spain, Springer-Verlag (Lecture Notes in ArtificialIntelligence, vol. 1415).

Experiential Learning Defined. (2015). In theThe University of Texas at Austin Learning Sciences. Retrieved from https://learningsciences.utexas.edu/teaching/engagement/experiential-learning/defined

Gascueña J., and Fernández-Caballero., (2005). *An Agent-Based Intelligent Tutoring System for Enhancing E-Learning/E-Teaching.*Retrieved from http://www.itdl.org/journal/nov\_05/article02.htm

Chan Lane H., (2006). *Intelligent Tutoring Systems: Prospects for Guided Practice and Efficient Learning*. Retrieved fromhttp://www.iaeng.org/publication/IMECS2012/IMECS2012\_pp63-67.pdf

Ghadirli H., and Rastgarpour M., (2012). A Model for an Intelligent and Adaptive Tutor based on Web by Jackson’s Learning Styles Profiler and Expert Systems.*Proceedings of the International MultiConference of Engineers and Computer Science, 1,* Retrieved from http://www.iaeng.org/publication/IMECS2012/IMECS2012\_pp63-67.pdf

Hill, Robin., (1998). *What Sample Size is "Enough" in Internet Survey Research* (pp. 3 – 4). New Zealand: Interpersonal Computing and Technology: An Electronic Journal for the21st Century

Java Tutorial. (2016). Retrieved from https://www.tutorialspoint.com/java/

Kaufer, M.J, Emoodji, (2015), GitHub repository, https://github.com/mjkaufer/Emoodji

Landowska A., (2013). *Affect-awareness Framework for Intelligent Tutoring Systems.* Retrieved from http://kio.eti.pg.gda.pl/publications/files/2013\_Landowska\_Affect-awareness\_Framework\_HSI.pdf

Lesson: Language Basics (2015). Retrieved from http://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html

Lewis, L.H. & Williams, C.J. (1994). In Jackson, L. & Caffarella, R.S. (Eds.). *ExperientialLearning: A New Approach* (pp. 5-16). San Francisco: Jossey-Bass.

Lim-Rañola et al., (2015). *An Agent-Based Intelligent Tutoring System for Java Using the Experiential Gamming Model as the Pedagogical Module*(Unpublished undergraduate dissertation). University of Santo Tomas, Manila, Philippines.

Malik, D.S. (2012). *Java Programming: From Problem Analysis to Program Design, Fifth Edition*. Boston, Massachusetts: Course Technology

Moon, J.A. (2004). *A Handbook of Reflective and Experiential Learning: Theory and Practice*. New York: RoutledgeFalmer.

Mcleod S., (2013).*Kolb – Learning Styles*. Retrieved from http://www.simplypsychology.org/learning-kolb.html

Mcleod S., (2012). *Experimental Method*. Retrieved from http://www.simplypsychology.org/experimental-method.html

Melis et al., (2001). ActiveMath: A Generic and Adaptive Web-Based LearningEnvironment. *International Journal of Artificial Intelligence in Education, 12,* 385-407.

Sampathkumaret al. (2014). Designing an Adaptive Distributed Tutoring System based on Students’ Learning Style and Collaborative Learning using Intelligent Agents. *International Journal of Computer Applications, 87(17),* Retrieved from http://research.ijcaonline.org/volume87/number17/pxc3893972.pdf

Schildt, H (2007). Java: The Complete Reference. Retrieved from http://iiti.ac.in/people/~tanimad/JavaTheCompleteReference.pdf

Tuaksubun C., and Mungsing S., (2007). *Design of an Intelligent Tutoring System that Comprises Individual Learning and Collaborative Problem-Solving Modules*. Retrieved from http://ijcim.th.org/SpecialEditions/v15nSP3/P17eLearningAP\_DesignOfAnIntelligent.pdf

Whiteley W., (2008). *Artificially Intelligent Adaptive Tutoring System*. Retrieved from http://cs229.stanford.edu/proj2008/Whiteley-ModernIntelligentTutoringSystem.pdf

Whitehill et al., (2008). *Automatic Facial Expression Recognition for Intelligent Tutoring Systems*. Retrieved from http://mplab.ucsd.edu/~jake/its08.pdf

Valerie S., and Psotka J., (1994). *Intelligent Tutoring Systems: Past Present and Future.* Retrieved from http://www.dtic.mil/dtic/tr/fulltext/u2/a280011.pdf

Appendix A

# Survey Questions

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

School: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Don’t try to think too hard, just answer honestly!**

*For Question 1*

The current setup of teaching (Experiential Learning) is presented by an *overview/introduction* (Doing/Reading, Thinking and, Concluding) followed by*video and practical coding* itself (Testing)

1. Do you think that the current setup is effective in teaching Java?
2. Yes
3. No
4. What was the most effective part in teaching?
5. The overview/introduction of the lesson
6. The video and practical coding in the exercise
7. Both
8. Do you think that the hints provided by the Facial Expression Recognition in the exam were useful?
9. Yes
10. No

*For Questions 4-6*

*Rate the following (1 being the lowest and 10 being the highest)*

1. The effectiveness of the overview/introduction of the lesson in teaching \_\_\_
2. The effectiveness of the video and practical coding in the exercise in teaching \_\_\_
3. The attractiveness of the user interface \_\_\_
4. Rate the module’s exercise in terms of difficulty:

1- Easy

2- Moderate

3- Difficult

4- Very difficult

\_\_\_ Basic Syntax

\_\_\_ Variables

\_\_\_ Basic Operators

\_\_\_ Conditionals

\_\_\_ Loops

\_\_\_ Methods

\_\_\_ Objects & Classes

1. Rate the module’s exam in terms of difficulty:

1- Easy

2- Moderate

3- Difficult

4- Very difficult

\_\_\_ Basic Syntax

\_\_\_ Variables

\_\_\_ Basic Operators

\_\_\_ Conditionals

\_\_\_ Loops

\_\_\_ Methods

\_\_\_ Objects & Classes

1. Rate the modules in terms of interest:

1- Not interesting

2- Slightly interesting

3- Neutral

4- Interesting

5- Very interesting

\_\_\_ Basic Syntax

\_\_\_ Variables

\_\_\_ Basic Operators

\_\_\_ Conditionals

\_\_\_ Loops

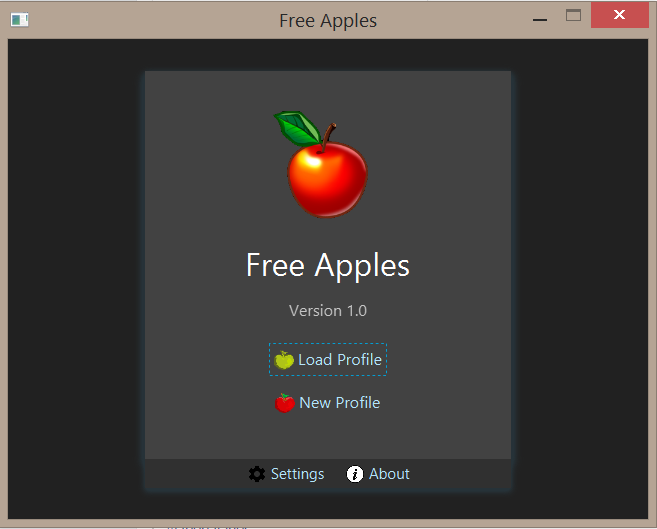
\_\_\_ Methods

\_\_\_ Objects & Classes

1. Do you find Java interesting after learning it?
2. Very interesting
3. Interesting
4. Neutral
5. Slightly interesting
6. Not interesting

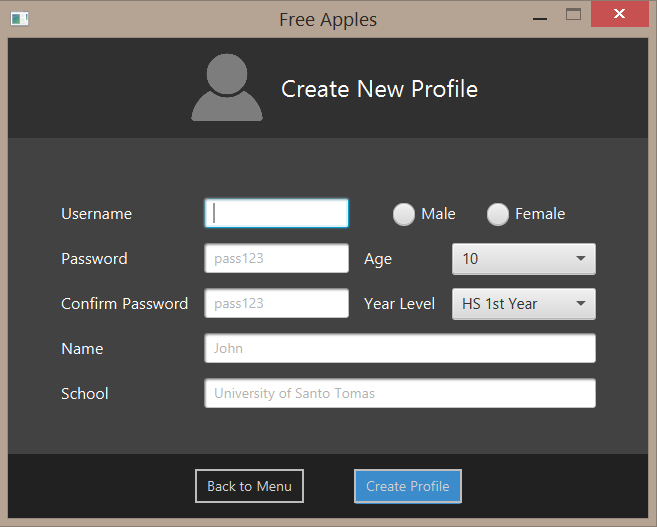
Appendix B

# User Interface



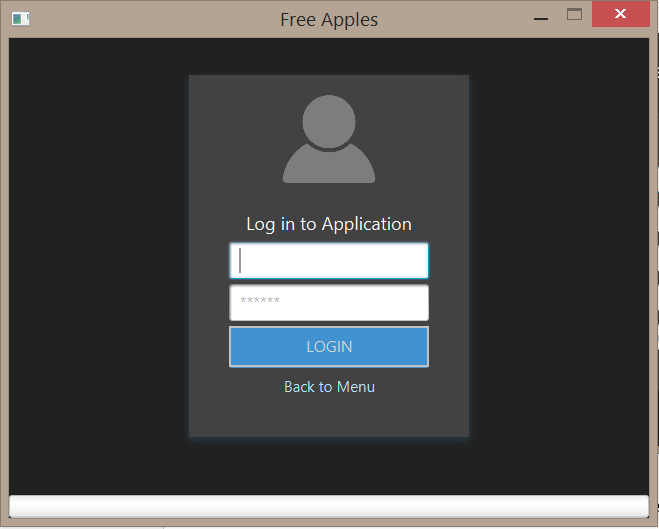
Start Menu Interface

* Allows the student to create or load profile
* Allows the student to access setting or about page



**Create Profile Interface**

* Allows the student to create a profile
* Allows the student to input his credentials



Login Interface

* Allows the student to access or log in his profile.



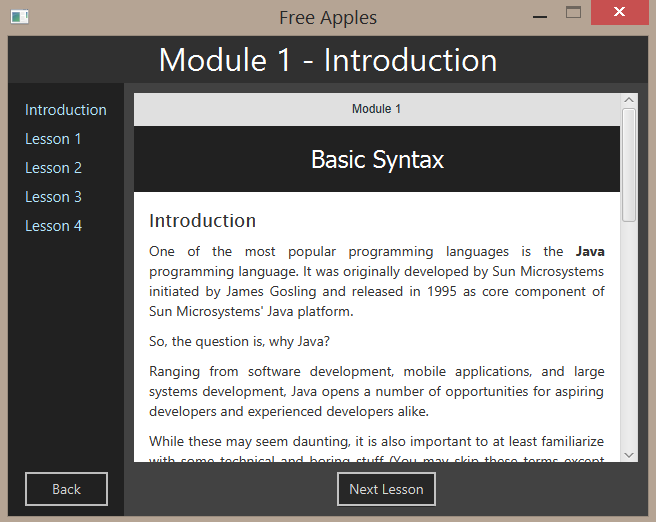
**Home Interface**

* Allows the student to access Modules through Learn.
* Allows the student to access Exam once he or she has finished a module.
* Allows the student to view his or her grades on exams.
* Allows the student to tackle challenges once all modules and exams are completed.

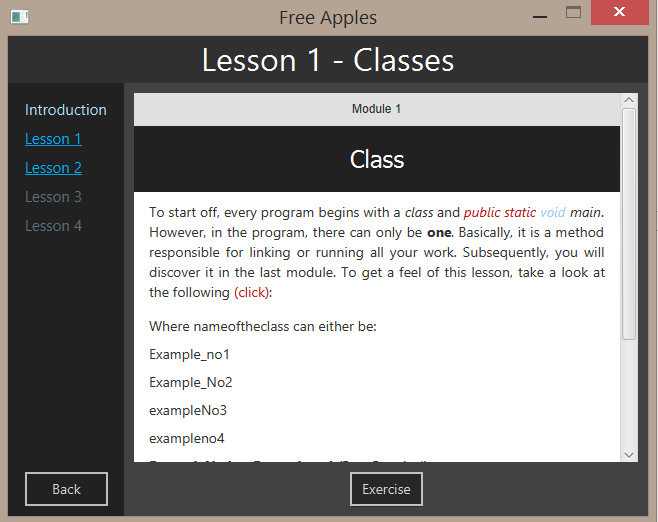


Learn Module Interface

* Allows the student to access available modules.

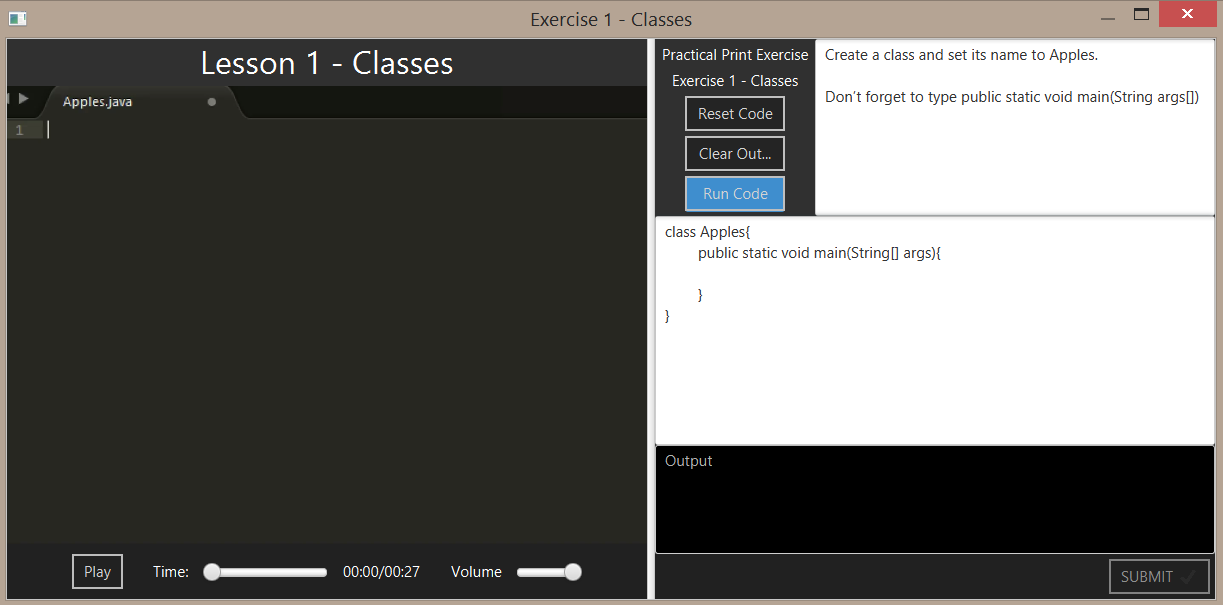


Introduction Module Interface



Lesson - 1 Module Interface

* On the left sidebar, these are the lessons available.
* Available lessons are highlighted meaning these are unlocked.
* Locked lessons are dull in color.
* Displayed on the right is the Text Lesson coded in HTML and JavaScript, also there are text/s that can be shown/hidden when clicked.

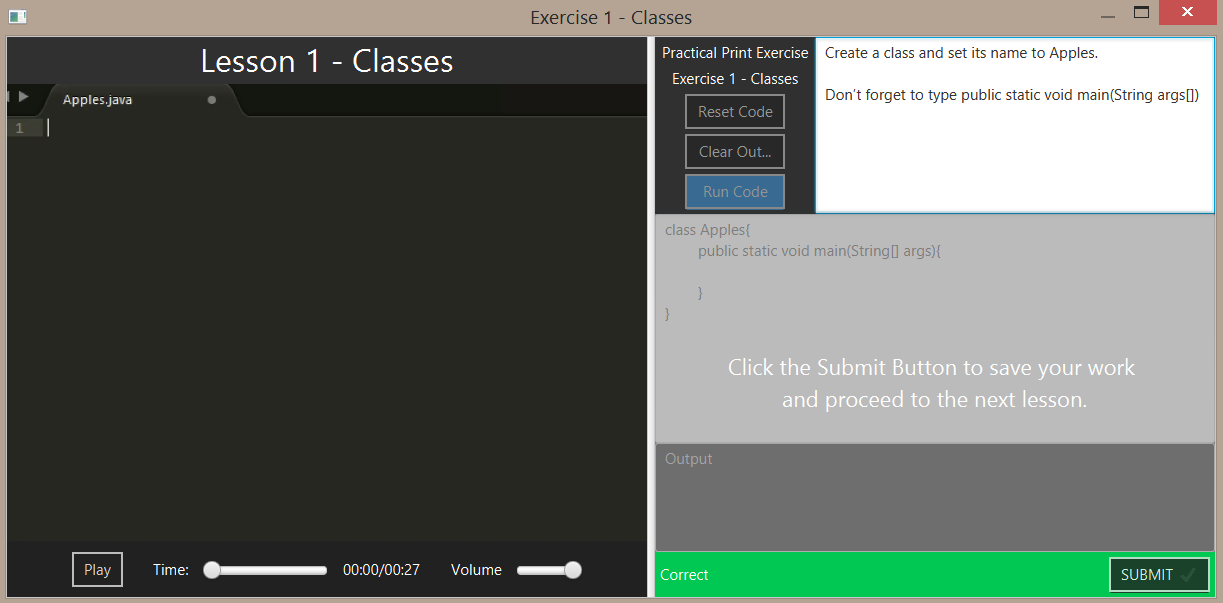


Exercise Interface

On the left portion is the Video Lesson. It contains a play button along with a time slider and a volume slider. For the right portion, it contains the Practical Exercise (text area) and instructions. Beside the instructions are the following:

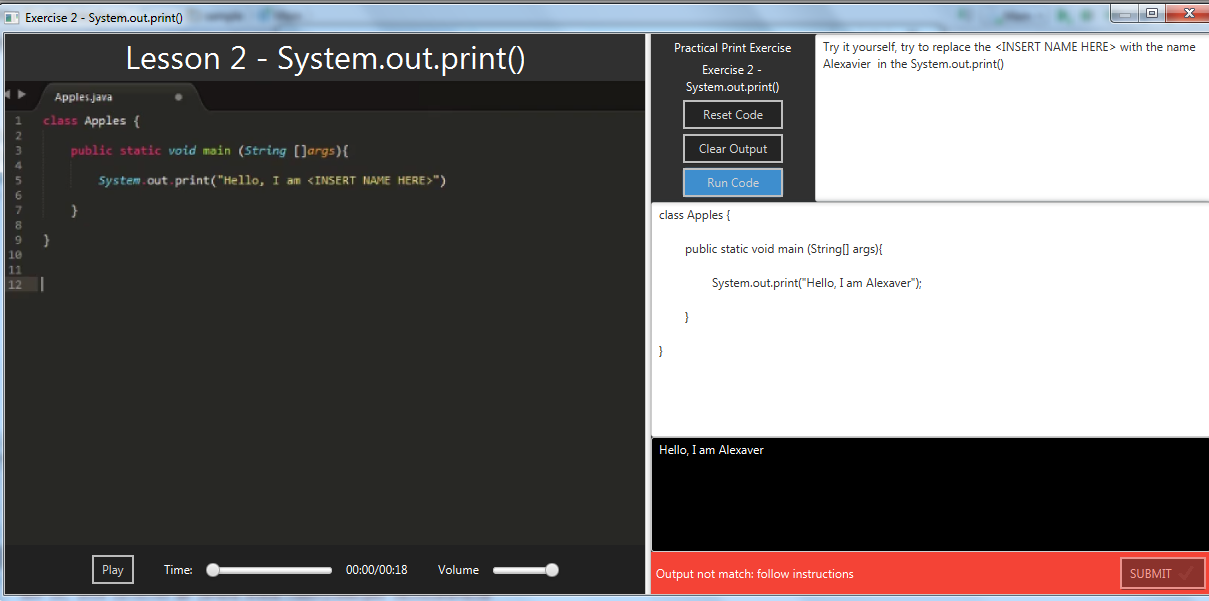
* Reset Code - to undo/reset the code inputted
* Clear Out - to clear the console or the output window
* Run Code- compiles the code

After successfully running the code, the submit button can be user to save the current code and proceed to the next lesson. Lastly, below the Practical Exercise is the console or output window wherein the output can be displayed.



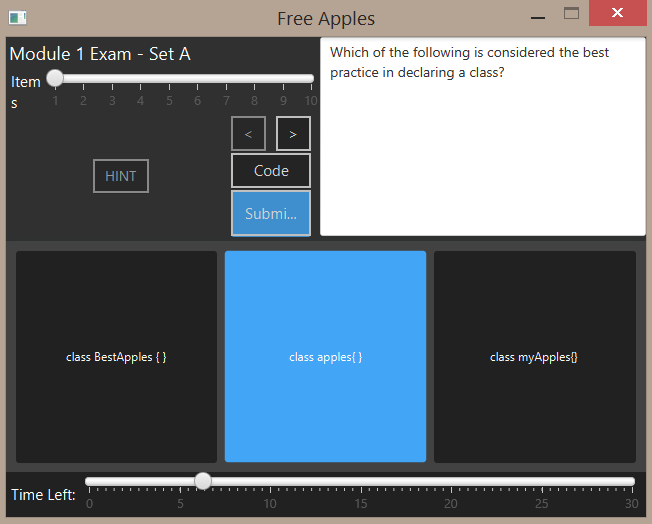
Exercise Interface After Correct Submission

* After correct submission, the Agent displays a “Correct” response.

****

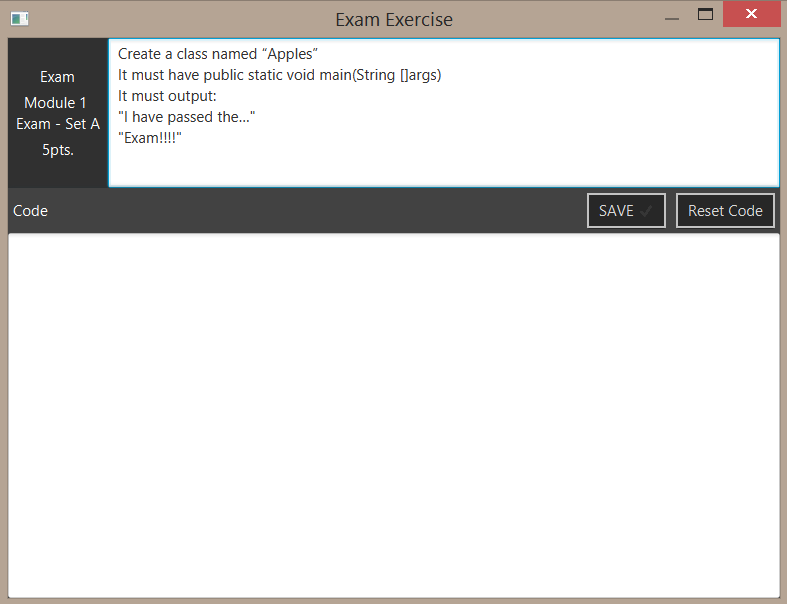
Exercise Interface After Incorrect Submission

* After incorrect submission, the Agent displays a response as a form of tip if the student did not follow instructions or a compilation error.



Exam Interface

* The upper left section of the window displays the number of items in a form of a slider.
* The left (<) and right (>) buttons can be used to proceed or backtrack an item.
* The hint button is locked, it is only unlocked if the agent detects an anxious expression.
* The upper right section of the window contains the question.
* The choices are shown and highlighted when selected.
* At the bottom is the time limit and the amount of time that has passed.



Exam Interface – Practical Coding

Similar to the exercise albeit without the feedback from the agent. There is an option of resetting the code in case the student intends to. If the student is finished, he or she can press the SAVE button in order to submit his or her code.

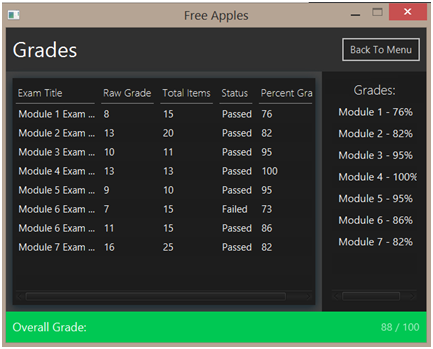
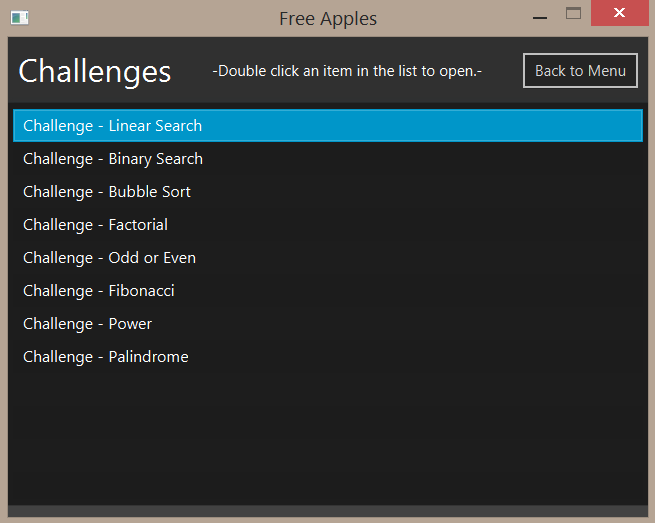


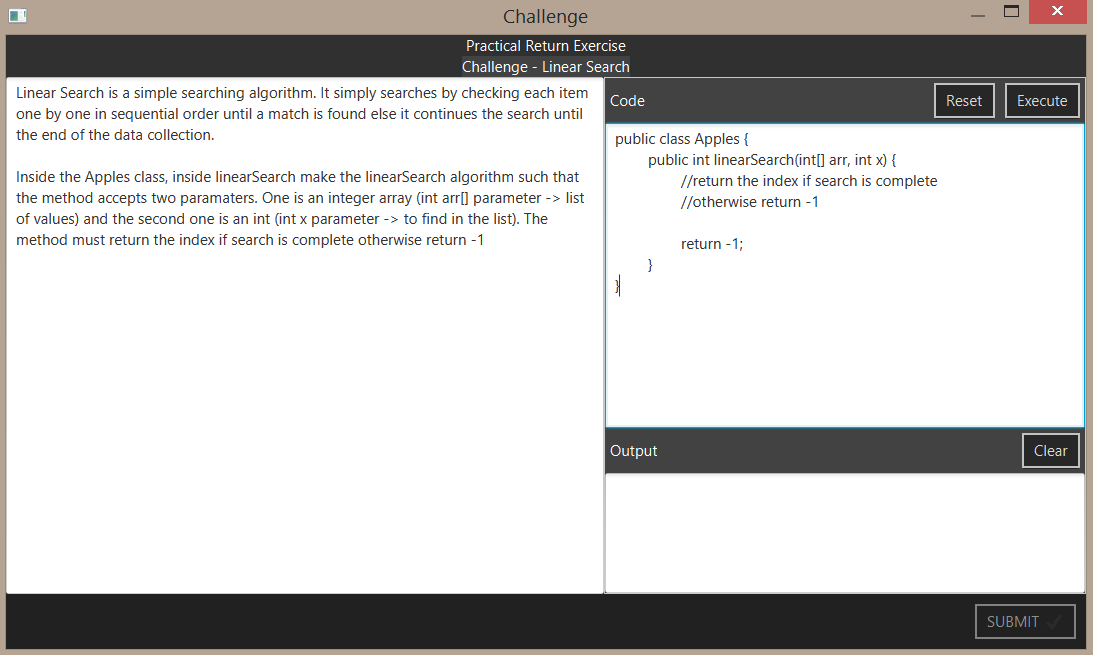
Figure IV - 31: Grades Interface

* Displays all the records of the student for each exam.
* It shows the raw grade, total items, status and percent grade.
* It also shows the grades of the student for each module and his overall grade.



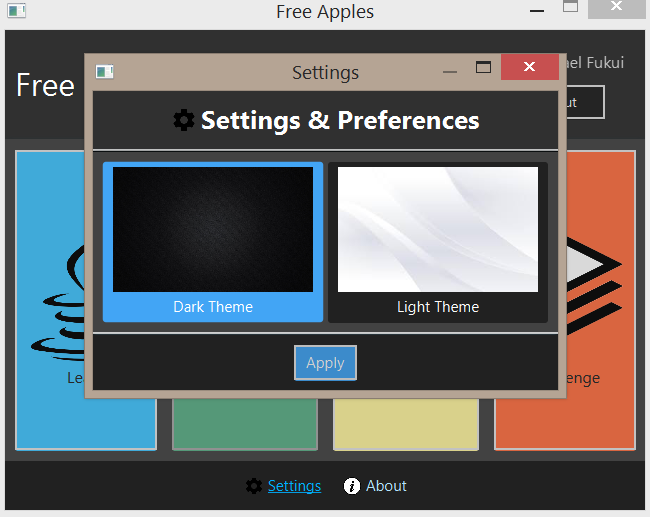
Challenges Interface

* Displays the list of all the challenges that the student may try.



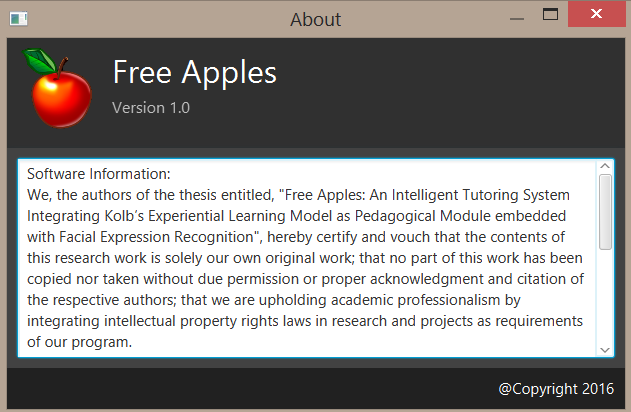
Challenges Interface –Practical Coding

* Left section of the window contains instruction/problem.
* Right section contains input (text area) for coding and output window



Settings Interface

* Allows the student to change the theme of the system.



About Page Interface

* About page displays the software information.

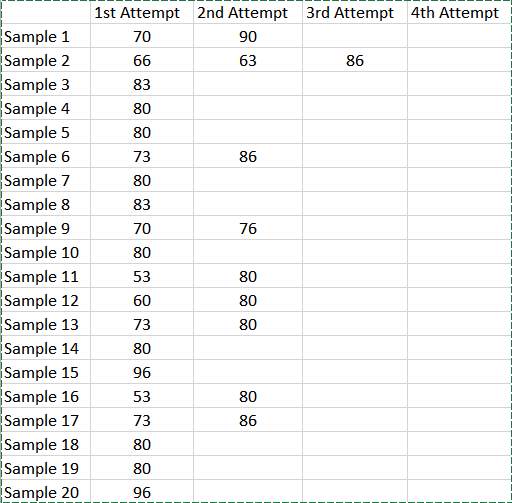
Appendix C

# Test Cases

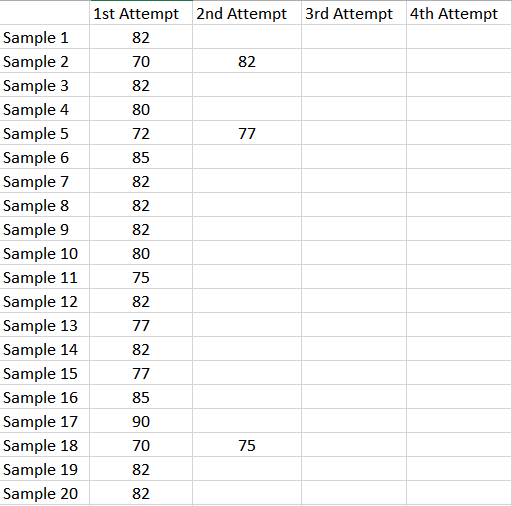
|  |  |  |  |
| --- | --- | --- | --- |
| **Scenario** | **Test Step** | **Expected Result** | **Actual outcome** |
| Onload page | | | |
| System Must Load Properly | Click or open the app then wait for it to open or load | Loads or Opens | Loads or Opens |
| Sign up page | Click the "Create new profile" button | "Create new profile" button is working and proceeds to sign up page | "Create new profile" button is working and proceeds to sign up page |
| Sign in page | Click the "Load profile" button | "Load profile" button is working and redirects to sign in page | "Load profile" button is working and redirects to sign in page |
| About page | Click the about icon button below | The about icon button is working and redirects to About page | The about icon button is working and redirects to About page |
| Settings page | Click the setting icon button | The setting icon button is working and redirects to settings page | The setting icon button is working and redirects to settings page |
| Create new profile page | | | |
| Validation of textfields | Enter random data including characters and non-matching passwords to see if there is validation | Validation alert occurs when not matched, incomplete, special characters, and other incorrect data inputs | Validation alert occurs when not matched, incomplete, special characters, and other incorrect data inputs |
| Data is pushed to database upon submit | Click the submit button after filling up all the fields | User data is pushed to database | User data is pushed to database |
| Load profile page | | | |
| Compare and validate username and passwords | Fill in the fields for username and password | Redirect to user page if valid, else produce an alert | Redirect to user page if valid, else produce an alert |
| User page | | | |
| Sign out | Click the sign out button | Cleans user session and redirects to Onload or guest page | Cleans user session and redirects to Onload or guest page |
| Locked pages | Check if the grades, challenges and exam pages are locked on first signed in by clicking through the buttons | Pages are locked and will not be opened until finishing the requirements | Pages are locked and will not be opened until finishing the requirements |
| Learn page unlocked | Check if the learn page is unlocked after first sign | Unlocked button | Unlocked button |
| Learn page | | | |
| Unlocked lessons | Click Module 1 and click Lesson 1 | Clickable and redirects to 1st lesson | Clickable and redirects to 1st lesson |
| Open exercise | Click the exercise button after loading the lesson | Open the practical exercise with videos included | Open the practical exercise with videos included |
| Unlock new lessons | Correctly answer the exercise on each lesson | If it is the last lesson , it unlocks a new lesson on the module and the exam | If it is the last lesson , it unlocks a new lesson on the module and the exam |
| Locked modules and lessons | Click to check | Does not open until the user passed the exams and exercises | Does not open until the user passed the exams and exercises |
| Video Practical Exercise (Lesson Exercises) page | | | |
| Instructions | Check the instructions in the textarea | No typo when it comes to names of the classes, methods, variables, etc. | No typo when it comes to names of the classes, methods, variables, etc. |
| Video | Click the play button in video | Video is working with no problem. All videos are correct and similar to instructions | There are videos with errors and does not match with the instructions |
| Code text area | Click and type to the text area | Able to focus and type | Able to focus and type |
| Exercise compilation error | Type a bunch of texts that the compiler cannot understand | Produce an alert or error message/ compilation error | Produce an alert or error message/ compilation error |
| Exercise incorrect output | Do not follow the instructions in the exercise | Produce an alert or error message stating the problem | Produce an alert or error message stating the problem |
| Cheat | Use tricks so that it may bypass the checking ex: instead of calculating and outputting the sum of 8 + 6 just output 14 in the console | Displays an error that indicates he or she did not follow the instruction | Displays an error that indicates he or she did not follow the instruction |
| Run to check | Click the run button | Compile and check for errors. If no errors, enable the submit button, or else produce the error | Compile and check for errors. If no errors, enable the submit button, or else produce the error |
| Submit disabled | Check if submit button is disabled while exercise is not yet finished | Submit button is disabled | Submit button is disabled |
| Submit enabled | Type the correct answer for the exercise, and then run code. Afterwards, try to click the submit button | Save the exercise, and then close the window | Save the exercise, and then close the window |
| Close | Click the close button | Alert will show that the exercise will not be saved | Alert will show that the exercise will not be saved |
| Exam page | | | |
| Timer and Timer slider | Check the timer in the exam window if it is moving | Timer slider is moving and once time has expired, it submits all answers | Timer slider is moving and once time has expired, it submits all answers |
| Left and right button | Click the left and right arrow buttons for navigating through the items of the exam | Can navigate from the first question upto the last one | Can navigate from the first question upto the last one |
| Questions and choices | Check by navigating each item | Correct set of questions and choices. | Correct set of questions and choices. |
| Choices button | Click to answer | Is clickable, and is marked as an answer | Is clickable, and is marked as an answer |
| Exercise | Click the code button to answer exercise | Opens up the practical print exercise | Opens up the practical print exercise |
| Submit button | Click the submit exam button | Opens the code exercise if not yet answered and time has not yet expired. Else, it computes the grade. | Opens the code exercise if not yet answered and time has not yet expired. Else, it computes the grade. |
| Close button | Click the close button | Closes the exam | Closes the whole application |
| Challenges page | Complete the last exam of module 7 | Unlocks the challenges page | Unlocks the challenges page |
| Grades page | | | |
| User grades | Check if the user grades is listed in the table | All grades in exams and retakes are there. Also there is an overall grade | All grades in exams and retakes are there. Also there is an overall grade |
| Challenges page | | | |
| List of challenges | Check if the page contains a list of all challenges | All challenges are listed in table with clickable links | All challenges are listed in table with clickable links |
| Challenge links | Click the links to answer the challenge | Opens the Practical Return Exercise | Opens the Practical Return Exercise |
| Practical Return Exercise (Challenges exercise) page | | | |
| Instructions | Check the instructions in the text area | No typo when it comes to names of the classes, methods, variables, etc. | No typo when it comes to names of the classes, methods, variables, etc. |
| Code text area | Click and type to the text area | Able to focus and type | Able to focus and type |
| Exercise compilation error | Type a bunch of texts that the compiler cannot understand | Produce an alert or error message /compilation error | Produce an alert or error message /compilation error |
| Exercise not match actual outputs with expected outputs | Try to return a value using different formula or just a dummy value | Produce an alert or error message stating the problem. The console displays the inputs, expected outputs, and actual outputs together with its match along with those who did not. | Produce an alert or error message stating the problem. The console displays the inputs, expected outputs, and actual outputs together with its match along with those who did not. |
| Every expected outputs matches with actual outputs | Follow the instructions and type the correct answer | Displays a message that all expected outputs and actual outputs matched. | Displays a message that all expected outputs and actual outputs matched. |
| Execute | Click the execute button | Compile and check for errors. If no errors, enable the submit button. Else, produce the error | Compile and check for errors. If no errors, enable the submit button. Else, produce the error |
| Close button | Click the close button | Closes the challenge page | Closes the challenge page |

**Appendix D**

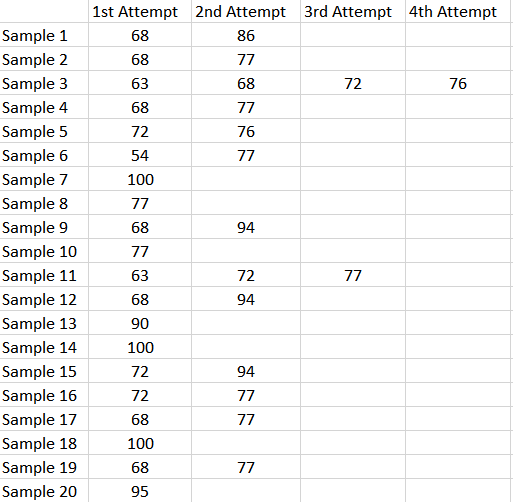
# Exam Results



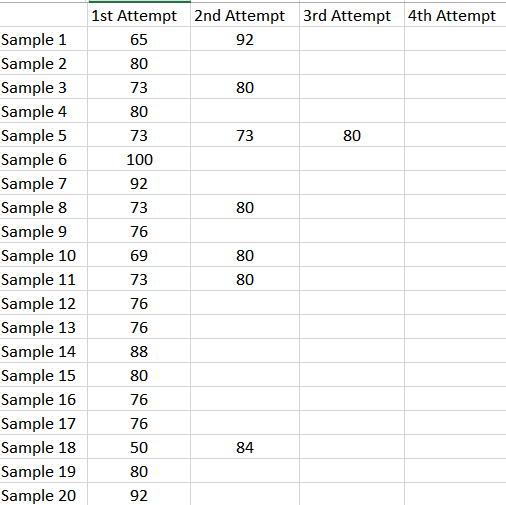
**Module I – Basic Syntax**



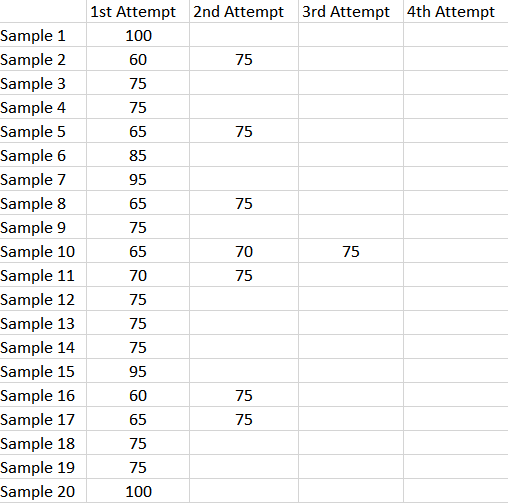
**Module II – Data Types and Variables**



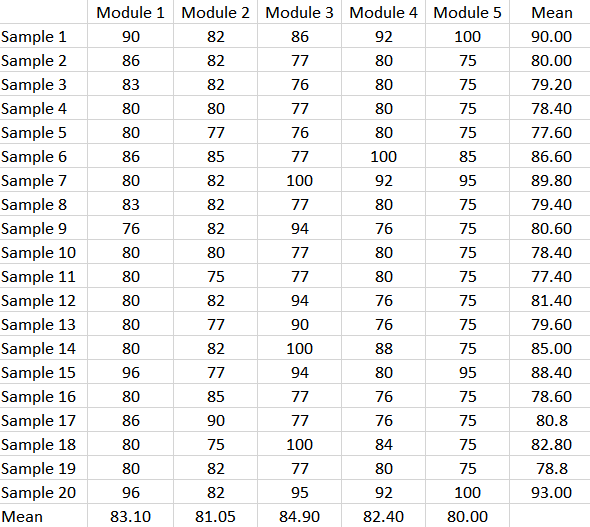
**Module III – Basic Operators**



**Module IV – Conditionals**



**Module V – Loops**



**Average exam grade per module**