

Diploma in Information Technology
Problem Solving
Instruction for CA3 Individual Assignment
October 2022 Semester

Assessment

100 marks (This assignment constitutes 40% of the overall assessment)

Deliverables

There are THREE (3) deliverables in this assignment, students must complete ALL components.

1. Written project report (10%)
 - a. Program design descriptions (2%)
 - b. Flowcharts (4%)
 - c. User guides with sample inputs and outputs (4%)
2. Python codes (20%)
3. Oral Presentation (10%)

The Task

The objective of this project is to allow students to design and implement a mini program. You are to use flowchart to show the design of your program, with clear steps and flows indicated. You need to use Python programming language to code all the programs, according to the project requirement.

In your report, write a short description, in 500 words, on your program design. You need to include the flowchart of your mini program in the report. You will also include screenshot samples on how the program should run in the report.

In your Python codes, include comments to explain the purpose of each sections.

Assessment Marks Allocation

| | Component Assessed | Marks Allocation |
|----------|--|-------------------------|
| 1 | Part 1a – Report | 5 |
| | Part 1b – Flowchart | 10 |
| | Part 1c – User guides with screenshots | 10 |
| 2 | Part 2a – Main Programs | 40 |
| | Part 2b – Additional / Intellectual Features | 10 |
| 3 | Part 3 – Oral Presentation | 25 |
| | Total | 100 |

The Case

CucKoo Learn designs and builds digital products for children and educators. The founders saw an opportunity in the rapid growth of online learning platform and decided to focus on creating software for interactive enhanced lessons. Currently *CucKoo Learn* is focused on education technology for primary school mathematics learning.

Targeted at children from Primary 1 to Primary 6, *CucKoo Learn* infuses interactive, media-rich animations and personalised technology to help children master Math skills on their own, all within a safe digital environment.

CucKoo Learn has been a partner and is being used by many primary schools for the past six years. The Math problems, challenges and games on *CucKoo Learn* are aligned with what children are learning in school, as they are based on the latest syllabus. *CucKoo Learn* currently has the largest user base of primary school students compared to other e-learning solutions.

To encourage self-motivation and to cater to different learning styles, such as visual, auditory and kinaesthetic, *CucKoo Learn* uses animated explanations to transform complex concepts into easy-to-understand visual instructions. Animated videos make learning more appealing, enabling children to quickly absorb knowledge, improve focus and remain engaged throughout the learning process.

To assess children learning progress, *CucKoo Learn* uses randomised quiz questions to test children understanding and ability in solving Math problems.

As an intern at *CucKoo Learn*, you are asked by your supervisor to write a mini program to automate simple arithmetic quiz creation (addition, subtraction, multiplication, division, etc) for Primary 1 students and to keep track on their results.

Project Requirements

After discussion with your supervisor, the following basic features are to be included in the program:

1. Register student
2. Start quiz
3. Check student's results
4. Display students' ranking

The program should display a menu to allow users to repeatedly perform the features above.

Details of the 4 features are as below:

1. Register student

To register/add a student into the program, the following data elements should be recorded:

- Name – the student's name
- Gender – Male or Female
- Class – the student's class

Advanced features to consider:

- To check if the student's name exists in the program
 - Register the student if it is not found
 - Display a message if the student is found

2. Start quiz

To start the quiz for a particular student, the following information should be recorded:

- The number of quizzes to practice
- The number of questions per quiz
- The types of questions
 - Addition; Subtraction; Multiplication; Division

The following process should be carried out:

- The program shall display a series of questions to the student
- Each question shall randomly generates two numbers, with an arithmetic operator
 - E.g. "What is 7 + 9?" or "7 + 9 ="
- The program shall allow the user to input the answer
- The program shall display a message to show whether the answer entered is correct or wrong

Advanced features to consider:

- To allow random types of questions, i.e. mixtures of addition, subtraction, multiplication, division, in a quiz.

3. Check student's results

To view the quizzes summary for a particular student.

Advanced features to consider:

- To report the quizzes statistics, e.g. percentages of correct answers

4. Display students' ranking

To show overall registered students ranking, based on the scores.

Advanced features to consider:

- To show class ranking.

Apart from the basic features indicated above, implementing additional and intellectual features, with good user interface, will earn you higher chance to be recruited as their full-time programmer.

Some additional features for consideration are:

- Loading students information and results from file
- Saving students information and results to file
- Input validations for secure programming
- Data analysis on students' performance
- Etc.

To show your ability in coding a mini program, your supervisor gave you the flexibility to design the structure and interface. Now, you shall apply what you have learnt in this module into practice.

Instructions

Cover Page

The cover page should include the institution name (and institution logo) the programme and the module name, the semester and year and date of submission. All these must be centralised in the page.

Write FULL Name and Student number as in the register on the cover. Students should a keep a copy of assignment submitted.

Python Codes

Suggested IDE and version: Wing Personal 8, Python 3.8 and above.

Please zip all your Python codes, together with the report, into one single file and upload it. If you used any additional Python library apart from the standard package, you need to include them in the submission.

Referencing

No referencing is needed for program designs and codes.

Font and Spacing

Font: Times New Roman

Font size: 12 and 1 ½ or double spacing.

Penalty Marks for Late Submission of Assignment

By one day: 20% to be deducted from total marks.

More than one day: submission will NOT be graded.

Plagiarism and Collusion

Students are not allowed to reuse old assignments or submit projects from previous semesters or copy largely from sources, particularly from the Internet web.

The submitted report and codes must show evidence that this is students' own work. No marks will be awarded if there are no reasonable explanations. Please be reminded that plagiarism and collusion is a serious offence, and all cases will be referred to the administration. Grades will be withheld if the submission is suspected of plagiarism or collusion till investigations are completed.

Important Dates of CA3 Report

CA3 Individual Assignment Deadline: 11th November 2022 11.59a.m.

Submit your project via Canvas, submission must be completed in order to be graded.

Lecturer Contact

You should contact your lecturer via your SIM email whenever you have any issue about your project. You may send your email to: chkoh005@mymail.sim.edu.sg

Marking Rubric

| <u>Criteria</u> | <u>Excellent</u> | <u>Very Good</u> | <u>Good</u> | <u>Acceptable</u> | <u>Weak</u> |
|-----------------------------------|---|--|--|---|--|
| <i>Project Report (6%)</i> | | | | | |
| <i>Description of the Program</i> | Student has provided detailed and step-by-step description of the program | Student has provided step-by-step description of the program | Student has provided sufficient description of the program | Student has provided limited description of the program | Student has provided short description of the program |
| <i>Appropriate Flowchart</i> | Student has used multiple appropriate flowcharts to describe the program completely | Student has used multiple appropriate flowcharts to describe the program | Student has used some flowcharts to describe the program | Student has attempted to use flowcharts to describe the program | Student has attempted to use flowchart to describe the program but were not sufficient |
| <i>Clarity of Presentation</i> | Student has provided clear, detailed | Student has provided clear and concise presentation | Student has provided clear presentation | Student has provided clear presentation to some extent | Student has attempted to provide a |

| Criteria | Excellent | Very Good | Good | Acceptable | Weak |
|---|---|--|--|--|---|
| | and concise presentation | | | | clear presentation |
| <i>Creativity of Presentation</i> | Student has used a variety of creative software and tools to present the assignment | Student has used many creative software and tools to present the assignment | Student has used some amount of creative software and tools to present the assignment | Student has used a couple of creative software and tools to present the assignment | Student has used at least one creative software or tools to present the assignment |
| User Guides (4%) | | | | | |
| <i>Screenshot Samples</i> | Student has provided clear, detailed and step-by-step screenshots guide on the program | Student has provided step-by-step screenshots guide on the program | Student has provided sufficient screenshots guide on the program | Student has provided limited screenshots guide on the program | Student has provided very little screenshots guide on the program |
| <i>Data</i> | Student has used appropriate and detailed test data on the program to illustrate the input and output | Student has used appropriate test data on the program to illustrate the input and output | Student has used sufficient test data on the program to illustrate the input and output | Student has used some test data on the program to illustrate the input and output | Student has used very limited test data on the program to illustrate the input and output |
| Python Codes (20%) | | | | | |
| <i>Understanding of the Project</i> | Student has demonstrated high level of understanding of the project requirements that are both stated in the task and those not mentioned | Student has demonstrated some understanding of the project requirements that are both stated in the task and those not mentioned | Student has demonstrated high level of understanding of the project requirements that are stated in the task | Student has demonstrated basic understanding of the project requirements that are stated in the task | Student has demonstrated sufficient understanding of the project requirements that are stated in the task |
| <i>Application of programming modules</i> | Student is able to apply majority of the topics learnt in this project | Student is able to apply many topics learnt in this project | Student is able to apply some topics learnt in this project | Student is able to apply at least two topics learnt in this project | Student is able to apply at least one topic learnt in this project |
| <i>Intellectual Features</i> | Student has demonstrated efficient use of intelligent algorithm in majority of the codes | Student has demonstrated the use of intelligent algorithm in most of the codes | Student has demonstrated some use of intelligent algorithm in the codes | Student has demonstrated basic use of intelligent algorithm in the codes | Student has demonstrated sufficient use of intelligent algorithm in the codes |

| Criteria | Excellent | Very Good | Good | Acceptable | Weak |
|---------------------------------|--|--|---|--|---|
| <i>Additional Features</i> | Student has added substantial number of additional features to the program | Student has added a good number of additional features to the program | Student has added some number of additional features to the program | Student has added a couple of additional features to the program | Student has added at least one additional feature to the program |
| <i>User Interface</i> | Student has created excellent user interface for the program | Student has created good user interface for the program | Student has created sufficient and simple user interface for the program | Student has created basic user interface for the program | Student has created poor user interface for the program |
| Oral Presentation (10%) | | | | | |
| <i>Clarity of Presentation</i> | Student has provided clear, detailed, and concise presentation | Student has provided clear and concise presentation | Student has provided clear presentation | Student has provided clear presentation to some extent | Student has attempted to provide a clear presentation |
| <i>Delivery Style</i> | Excellent control of volume, pace and diction; no distracting gestures; visual aids used effectively | Good volume and energy; good pace and diction; few or no distracting gestures; visual aids used adequately | Average volume and energy; generally good pace and diction; some distracting gestures or posture; visual aids could be improved | More volume/energy needed at times; pace too slow or fast; some distracting gestures or posture; visual aids could be improved | Low volume or energy; pace too slow or fast; poor diction; distracting gestures or posture; visual aids poorly used |
| <i>Content</i> | All information was relevant and appropriate to requirements of the assignment | Most information was relevant and appropriate to requirements of the assignment | Most information relevant; some topics needed expansion or shortened | Information was valid but not explicitly related to the assignment | Information was not relevant or directly related to the assignment |
| <i>Q & A Session</i> | Answers questions with authority and accuracy | Provides good, clear answers to questions | Provides adequate answers to questions | Some difficulty answering questions | Uneasiness or inability to answer questions |
| <i>Overall Impression Grade</i> | Student was very well prepared for the entire presentation | Student was well prepared for the most parts of the presentation | Student was prepared for most parts of the presentation | Student appeared prepared for some parts of the presentation | Student was clearly unprepared for the presentation |