



FACULTY OF ENGINEERING & COMPUTING

PROGRAMME : Foundation in Engineering, Science and Technology
ACADEMIC YEAR : 2019
MODULE : Structured Programming
MODULE CODE : FEST 027
MODULE LEADER : Mr Choo Han Lim
ASSIGNMENT TYPE : Report
INTAKE/GROUP : FEST 42
DISTRIBUTION DATE : Friday, 13 December 2019
SUBMISSION DATE : Friday, 24 January 2020 12:00 PM

STUDENT NAME	STUDENT ID	CLASS CODE
		FEST 42

Assignment Feedback Form

Structured Programming

<div>/100</div>	Mark
SYSTEM REQUIREMENT & CODE	/ 45
INPUT, CALCULATIONS & OUTPUT	/ 20
REPORT	/ 25
DEMO	/ 10

General Comments:

Assessor's Signature:	Date:
Name: Choo Han Lim	

General Instructions

Use the following format for the preparation of the *assignment submission*.

- Paper size : A4
- Margins: left, right, top and bottom = 2.54cm
- Font size : 12 , Times New Roman
- Line spacing : 1.5
- Alignment: Justify
- Number all pages sequentially (Middle, Bottom)
- Number all Figures and Tables sequentially and refer them in the text
- Binding: **staple at top left corner** of assignment submission

WARNING

- Assignments submitted after the due date will be considered late.
- Assignments submitted not later than two weeks after the due date will be marked, but the marks will be capped to a maximum of 40%.
- Assignments submitted later than two weeks will be marked, but carry zero mark.
- First City University College takes allegations of plagiarism very seriously. Submissions involving plagiarism will be marked, but given zero mark. Plagiarism is the attempt to pass off the work of another as your own. Information taken from the work of others should be acknowledged by reference to obviate the charge of copying.
- Collusion is an academic irregularity within the First City University College assessment regulations. Any student found colluding in the production of any assessment will be subject to an investigation with the imposition of any penalty deemed appropriate. Students must ensure they are familiar with the definition of collusion.

Note:

Plagiarized work will result in a 'FAIL GRADE'

I. Assignment Requirements

Introduction

Write a C program to conduct a **depression assessment**. The test allows the user to have an earlier detection of depression.

The program should be designed neatly and effectively. **Selection, repetition, array and 3 functions** must be used in the implementing the program.

The test questions must be based on any available personality test resources such as books, websites and others.

Following are the requirements of the program:

- Your program must prompt the user for their details (name, age, etc). You can use these details to make the test more personalized for the user.
- Your program must display a series of questions with various answer options (AT LEAST 4 options) for most of the question. You should have AT LEAST 20 questions in your test.
- Your program must prompt the user to respond by entering an answer from the options given.
- At the end of the test, calculate and display the test score and provide an analysis about the score.
- Your program must allow the user to choose to take the test again if they wish to. Your program should end when the user chooses to quit.
- You are required to include input validation in the program to validate user's inputs.

Submission Requirements

1. Hardcopy of report that contains the following:
 - Table of content with page number
 - Overview of the program
 - Flowchart of the program
 - Testing of the program (with screen shots as evidence and explanation for the result of the testing)
 - Reference
2. A publishable working copy of a compiled version of the completed assignment (.c) together with the softcopy of the report - both are to be uploaded to ELP.

II. Learning Outcomes

Learning Outcomes	Assessment
1. Identify and use fundamental program structures and methodology;	
2. Identify and use primitive data types and differentiate when each type should be used;	
3. Construct solution using flow-charting and/or pseudocode;	✓
4. Apply program constructs involving decision structures, loops , functions array and files;	✓
5. Work with an integrated development environment (compiler, linker and debugger) within a standard Operating system.	✓

III. Assessment Marking Criteria

Area of concern	Excellent 70 – 100 marks (A)	Good 60 -69 marks (B)	Satisfactory 50 - 59 marks (C)	Marginal 40 -49 marks (D)	Poor 0 – 39 marks (F)
System Requirements & Code (45M)	Program fulfils all the requirements and works correctly. Code is understandable, well-commented, and well-organized.	Program fulfils most of the requirements. Minor issues with consistent indentation, use of whitespace, naming, or general organization. Few places that could benefit from comments are missing them or the code is <i>overly</i> commented.	Program fulfils half of the requirements. At least one major issue with indentation, whitespace, names, or organization. Complicated lines or sections of code uncommented or lacking meaningful comments.	Program only fulfil a small portion of the requirement. Major problems with at least three or four of the readability subcategories. Comments are sparse.	Program does not fulfil any of the requirement. Program code does not work at all.
Input, Calculations & Output (20M)	User friendly and concise input and output. Score calculation are correct. Comprehensive analysis about the score are provided.	User friendly input and output but contains a minor issues with consistent use of whitespace Score calculation are correct. Analysis about the score are provided.	Quite user friendly. At least one major issues with instructions and error message, and use of whitespace. Score calculation are correct for most of the questions. Brief analysis about the score are provided.	Some instructions / error messages are missing. Minor issues with the score calculation and analysis.	Instructions / error messages are missing. Score calculation are incorrect. Analysis about the score are not provided.
Report (25M)	Detailed test case and correct flowchart. IEEE referencing is used. Screenshots are described in detailed.	Good but incomplete test case and correct flowchart. IEEE referencing are used with minor errors. Some of the screenshots are not properly explained.	Good but incomplete test case and minor flaw in flowchart. IEEE referencing are used with major errors. Explanation of screenshots are too brief.	A lot of test cases are not included and major flaw in flowchart. IEEE referencing are used with major errors. Screenshot are not described.	No test case and flowchart is included. References are not documented. No screenshot are provided.
Demo (10M)	Showed very good understanding. Able to explain the ins and out of the system and the logic behind it. Able to answer all questions that were asked.	Showed good understanding. Able to explain most of the logic behind the system. Able to answer most of the questions that were asked.	Showed some understanding. Able to explain some of the logic behind the system. Able to answer some of the questions that were asked.	Not quite able to explain the logic behind the system because of lack of understanding. Not quite able to answer the questions that were asked.	Can't explain the logic behind the system because lack of understanding & effort. Not able to answer the questions that were asked.