|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **ASSIGNMENT COVER PAGE** | | | C:\Users\hoching.tay\Desktop\Lincoln_UK_06092017-01.png |
| **Programme** | | | **Course Code and Title** | | |
| Bachelor of Computer Science (Hons)/  Bachelor of Computer Science (Hons) In Computer & Network Technology/  Bachelor of Software Engineering (Hons) | | | CPR3113/N Principles of Programming | | |
| **Student’s name / student’s id** | | | **Lecturer’s name** | | |
|  | | | Tan Phit Huan | | |
| **Date issued** | **Submission Deadline** | | | **Indicative Weighting** | |
| Week 3 - 26/09/2022 | Week 7 – 28/10/2022 | | | 30% | |
| **Assignment 1 title** | Selection and Iteration | | | | |
| This assessment assesses the following course learning outcomes | | | | | |
| **# as in Course Guide** | **UOWM KDU Penang University College Learning Outcome** | | | | |
| **LO1** | **Analyze algorithms to solve basic computing problems using flow charts and pseudocodes.** | | | | |
| **LO2** | **Demonstrate a computational solution using principle of selection and iteration.** | | | | |
| **# as in Course Guide** | **University of Lincoln Learning Outcome** | | | | |
| **LO3** | **Demonstrate the ability to select from a range of possible options, to provide justification for that selection, and to implement algorithms in a particular context** | | | | |
| **LO1** | **Characterise a problem in the context of possible solution mechanisms** | | | | |
| **LO2** | **Model a problem solution using appropriate vocabulary** | | | | |
| **LO1** | **Implement control flow with decisions and loops using good programming practices** | | | | |
| **LO2** | **Determine an appropriate algorithmic approach to a problem** | | | | |
| **LO2** | **Implement control flow with decisions and loops using good programming practices** | | | | |
| **LO3** | **Apply object-oriented principles to the implementation of software programs** | | | | |
| **Student’s declaration** | | | | | |
| I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.  Student’s signature: Submission Date: | | | | | |

|  |  |
| --- | --- |
| **Dates and Mechanisms for Assessment Submission and Feedback** | |
| **Mechanism for handout to students** | OpenLearning LMS |
| **Mechanism for submission of work by student** | *Soft copy online submission via OpenLearning* |
| **Date by which work, feedback and marks will be returned to students** | 11th November 2022 |
| **Mechanism for return of assignment work, feedback and marks to students** | Feedback will be provided by a marking template. This will be available to students via Open Learning. The discussions at the walkthroughs will also provide informal feedback |

# COURSEWORK SUBMISSION GENERAL INFORMATION

# Academic integrity statement

You must adhere to the university college regulations on academic conduct. Formal inquiry proceedings will be instigated if there is any suspicion of plagiarism or any other form of misconduct in your work. Students must **NOT** collude with other groups of students or plagiarise their work.

# Nature of the submission required

A soft copy of your assignment in **PDF version** should be submitted to lecturer, no later than the date and time stipulated on the cover sheet. In addition, an electronic copy of your work must be submitted to Turnitin. The first page of your report, immediately after the cover page, must be a page from Turnitin clearly showing your name and your Originality Score (Please refer to [submission arrangement](#_Submission_arrangement)).

Diagrams may be used where they are helpful to support your arguments or description. If they are not your own work, the source must be referenced. Please help us to handle and mark your work efficiently.

Please take note for group submission, only **one submission per group**. This will contain both the group and individual elements. The individual element must be clearly labelled to indicate which group member completed the task.

# Documentation guidelines

Student is required to submit a **SOFT COPY** of the report and ensure that it uses the following formatted styles: 1) Font type: **ARIAL**, 2) Font size: **11** **pt**., 3) Line spacing: **Single spacing** and 4) Page layouts: **Justify**. Please make sure you have proper format alignment for all paragraphs, following standard writing style and use **HARVARD CITATION STYLE** for citation. Please include a **HEADER** with the following information: **Student ID, Student name, Course code and Assignment type**. Please also include a proper cover page for your submission which contains information about the students, assignment, course, and department with UOW Malaysia KDU Penang University College and University of Lincoln (UoL) logos on top. Also include page number and a list of references, which is shown on the last page.

# Penalties for late submission

For late submission of this Assignment, a penalty of a reduction by 10% of the maximum mark may be applicable for each Calendar Day or part thereof that the submission is late. An Assignment submitted more than **TEN** Calendar Days after the deadline will have a mark of zero recorded for this Assignment.

# Submission arrangement

1. Cover page
2. Table of Content
3. Main Report
4. References or Bibliography (whichever applicable)

# Assignment instructions/Background

**Scenario:**

A healthcare center would like to have a system to handle the patients’ records. The administrator would need to use the system to key in the record of patients’ details (name, identity card number, height, weight, and blood type). Allow the administrator to repeat the process for any invalid input. The system will identify the patient’s gender and date of birth according to the identity card number. The format for the Malaysian identity card is set as **YYMMDDPB###G**, where the first 6 digits, **YYMMDD** indicates the date of birth (year-month-day), **PB** represents the place of birth, and the last digit **G** represents the gender (odd numbers represent male and even numbers represent female).

The system also calculates the body mass index (BMI) of patient according to the height and weight. The formula to calculate BMI will be weight (kg) / [height (m)]2. Then, the system will show the patient’s information (name, identity card number, height, weight, blood type, date of birth, gender, BMI, and weight status). The weight status will be output (see Table 1) according to the BMI based on the weight and height entered by the administrator. Your program should have a loop and continue to ask for the patient’s details until indicated otherwise.

**Table 1 BMI and weight status**

|  |  |
| --- | --- |
| BMI | Weight Status |
| 30.0 and Above | Obese |
| 25.0 – 29.9 | Overweight |
| 18.5 – 24.9 | Healthy Weight |
| Below 18.5 | Underweight |

You are free to design the interface or menu design.

You are free to design the interface or menu design.

**Question 1 (40%):**

## You are required to perform 3 tasks.

## Task 1: Defining diagram (10%)

Create a defining diagram to divide the problem into 3 different components (Input, Output & Processing).

## Task 2: Solution algorithm (20%)

Design a solution algorithm using flow-charts or pseudocodes based on the preceding defining diagram.

## Task 3: Checking the solution algorithm (10%)

Perform desk checking with 2 sets of input test data.

**Question 2 (60%):**

## Program implementation (60%)

Convert your solution algorithm to a complete Java program.

**Handing in your work**

Main report should contain the following:

* Defining tables, pseudocode/flow charts, and desk checking tables.
* Java source code.
* Description of the program with some screenshots of the running program and explanation.

**This is an individual assignment. *Each student*** *should upload the document in PDF and Java source code (.java files) in a zip file to the link provided in OpenLearning.*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CPR3113/N Principles of Programming**  **MARKING RUBRIC**  **ASSIGNMENT 1**  **Selection and Iteration (30%)** | | | | | | | | | | |
| **Question 1 (40%)** | | | | | | | | | | |
| **LEARNING OUTCOME** | **MARKING CRITERIA** | **SCALE** | | | | | | | | |
| **Fail**  **(0-49)** | **3rd Class**  **(50-59)** | **2nd Lower Class**  **(60-69)** | **2nd Upper Class**  **(70-79)** | **1st Class**  **(80-100)** | **YOUR MARKS/COMMENTS** | | |
| 100% | Weightage | Actual Marks |
| **CLO1** | **Task 1:**  **Defining Diagram**  **(10%)** | No defining diagram or only demonstrate a defining diagram with 1%-49% of program requirements achieved. | Demonstrate a defining diagram with 50%-59% of program requirements achieved. | Demonstrate a defining diagram with 60%-69% of program requirements achieved. | Demonstrate a defining diagram that covers 70%-79% of program requirements. | Demonstrates mastery in the use of defining diagram and achieved 80%-100% of program requirements. |  | 0.10 |  |
| **Task 2:**  **Algorithm Design**  **(20%)** | Demonstrate use of flow chart/ pseudocode 0%-49% of program requirements achieved. | Demonstrate use of flow chart/ pseudocode 50%-59% of program requirements achieved. | Demonstrate use of flow chart/ pseudocode with 60%-69% of program requirements achieved. | Demonstrates proficiency in use of flow chart/ pseudocode that covers 70%-79% of program requirements. | Demonstrates mastery in the use of flow chart/ pseudocode and achieved 80%-100% of program requirements. |  | 0.20 |  |
| **Task 3:**  **Desk Checking**  **(10%)** | Demonstrate desk checking with 0%-49% of program requirements achieved. | Demonstrate desk checking with 50%-59% of program requirements achieved. | Demonstrate desk checking with 60%-69% of program requirements achieved. | Demonstrate desk checking that covers 70%-79% of program requirements. | Demonstrates mastery in desk checking and achieved 80%-100% of program requirements. |  | 0.10 |  |
|  | **Total (40%)** | | | | | |  | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Question 2 (60 %)** | | | | | | | | | | | | | | |
|  | **MARKING CRITERIA** | **SCALE** | | | | | | | | | | | | | |
|  | **Fail**  **(0-49)** | | **3rd Class**  **(50-59)** | **2nd Lower Class**  **(60-69)** | **2nd Upper Class**  **(70-79)** | **1st Class**  **(80-100)** | | **YOUR MARKS/COMMENTS** | | | | | | |
| 100% | | | | | Weightage | Actual Marks |
| **CLO2** | **Code Quality**  **(20%)** | Very poor coding which is hard to understand. Little use of comments. Poor naming of almost all classes, methods and variables. | | A poor attempt; which may be several problems with structure, or very little use has been made of comments, or the naming of classes, methods and variables are unsatisfactory in a significant number of cases. | A fair attempt; the code is in reasonable quality with several omissions of naming and use of comments. | Generally, a good attempt, making use of comments, and where the majority of classes, variables and methods have been appropriately named. However there may be several omissions of Javadoc comments and the code. | Good use of commenting throughout, including Javadoc comments for the vast majority of classes, methods and variables. | |  | | | | | 0.20 |  |
| **Use of appropriate programming concepts (decision/repetition)**  **(20%)** | Does not demonstrate any logical use of appropriate programming concepts. | | Demonstrate some limited use of appropriate programming concepts. | Demonstrate use of appropriate programming concepts. | Demonstrates proficiency in the use of appropriate programming concepts. | Demonstrates mastery in the use of appropriate programming concepts. | |  | | | | | 0.20 |  |
| **Program execution and output quality**  **(User friendliness, Error free during runtime)**  **(20%)** | Program does not able to compile. | | Program executed with runtime error but achieve partial program requirements. | Program executed error free with limitations to achieve minimum program requirements. | Program executed error free with correct output and achieve all program requirements. | Program executed error free with excellent output with appropriate validation. | |  | | | | | 0.20 |  |
| **Total (60%)** | | | | | | | | |  | | | | | | |
|  |  | |  | | | | |  | |  |  |  |  | | |
| **Overall score (100%)** | | | | | | | | |  | | | | | | |