Dublin City University
National Institute for Digital Learning
Open Education

Module: C2: Object Oriented Programming with Java

Programme(s): BSc in Information Technology
BSc in Management of Info Tech/Info Syst
Certificate in Computer Programming
HDIP in Software Development

Assignment 2 2020–2021

Question 1 **(25 marks)**

Write a Java program that displays six random colours of the rainbow and then prompts the user to enter the missing colour of the rainbow. If the user guesses correctly the program should display “Correct Answer!”, and exit. If the user types in an incorrect answer, the program should inform the user of the correct answer and start again with a new set of six random colours, from which the user is to guess the missing colour.

The program should store the colours of the rainbow in an array, and use a for loop to iterate through the array when displaying the colours on screen.

Example program execution (user input is shown in italics):

```
What colour of the rainbow is missing from this list?
red
orange
yellow
green
indigo
violet

blue
Correct Answer!
```

Question 2 **(25 Marks)**

Write a program that prompts the user to enter a character in the range of A-E (inclusive), and four integers in the range of 1-9 (inclusive). The program should compute the total by multiplying the numeric equivalent of the character input (where A is 1, B is 2, C is 3, D is 4, E is 5) by the sum of the integer inputs.

The program should then print to a file as follows:

- The first line in the file should contain headings as shown in the example file contents below. Each heading should be separated by |.
- The second line should contain the user inputs and the total. Each value should be separated by |.
- The file name should be of the format YYYYMMDD.dat
  - YYYYMMDD is today’s date in year month day format
  - .dat is the filename extension.
  - Example filename: 20210131.dat.

Example file contents:

```
char|int1|int2|int3|int4|total
B|2|4|3|6|30
```

Question 3 (25 Marks)

A customer counter is used to admit a limited number of people to a venue. For example: as a person enters a venue, a button is clicked to increase the count. As someone leaves, another button is clicked to reduce the count.

Write a CustomerCounter class to implement this behaviour. Include sample code in main() that creates a CustomerCounter object and simulates some random entry and exit of people that causes the maximum limit of customers to be reached and cleared.

If the button for a person entering was clicked at the allowable limit, simulate an alarm by printing a message “Alarm: Maximum number of *x* customers reached, please ask next customer to wait”, where *x* is the maximum number of people allowed. If a customer leaves when it was at the maximum capacity, give a message to indicate it is okay to let another customer in.

The CustomerCounter class should be in its own separate .java file.
The sample program with the main() method in another.

Question 4 (25 Marks)

Create a Vessel superclass, and two classes (Boat and Dinghy) that inherit from Vessel. A vessel has a name, length, make and year of manufacture. A Boat has a steering type (one of these: *WHEEL, TILLER, OARS*), and a Dinghy has a boolean indicating if it is inflatable. The Vessel class has a toString method which returns a string listing all of the details it has. The Boat and Dinghy override this method to extend it with their details.

Write the class declarations with the member variables, the constructors, getters and setters, and the methods toString for all classes. Supply a test program that tests these classes and methods.

Deliverables:

A zip file containing:

For Question 1: A program called c2*a2**q1**_2021_StudentName.java*

For Question 2: A program called c*2a2**q2**_2021_StudentName.java*

For Question 3: A program called c*2a2**q3**_2021_StudentName.java*
A class called *CustomerCounter.java*

For Question 4: A program called c2*a2**q4**_2021_StudentName.java*
And also 3 separate files for each class, as follows:
Vessel.java
Boat.java
Dinghy.java

- **All programs must**

➢ INCLUDE COMMENTS

➢ INCLUDE CODE THAT IS CORRECTLY ALIGNED AND INDENTED

➢ COMPILE WITH NO ERRORS

➢ RUN WITHOUT CRASHING

BEFORE SUBMITTING AN ASSIGNMENT, PLEASE FAMILIARISE YOURSELF WITH THE ASSIGNMENT REGULATIONS. THESE ARE AVAILABLE AS APPENDIX 3 OF THE COURSE HANDBOOK 2020–2021. (The Course Handbook is available here: http://moodle.dcu.ie/mod/resource/view.php?id=254230)

YOU SHOULD ALSO ENSURE THAT YOUR ASSIGNMENT FILES ARE CORRECTLY NAMED (SEE SECTION ON *Saving and Naming Your Online Assignment File* IN THE ASSIGNMENT REGULATIONS) AND THAT YOUR FILES ARE IN A COMPATIBLE FORMAT (SEE APPENDIX 10 OF THE COURSE HANDBOOK 2020–2021).