**THE HANG SENG UNIVERSITY OF HONG KONG**

**Bachelor Degree Programmes**

**Midterm Exam**

**Module Code: COM1102**

**Module Title: Programming and Data Structures**

**Time: 90 minutes**

**Some questions may contain part marks even if you cannot solve it completely.   
Do not over worry if you cannot finish all problems. (Especially if you are a “slow” programmer)  
You are suggested to read all questions quickly first before you start. Good luck!**

**Notes to Candidates:**

1. This is an open book test.
2. There are 80 marks in total.
3. Save your works frequently.
4. Submit the required files (e.g., this question paper, your source codes and project zip files) to Moodle before the deadline.
5. This paper must be your individual effort. Any form of help, collaborations, discussions, consultations, or cooperations with other candidates, or anyone both inside or outside HSUHK, whether online or offline, are strictly forbidden.
6. No two students may take the test at the same location unless with prior approval.
7. Camara must be turned on for showing yourself and your immediate neighbourhood while working on the test.
8. Contact your instructor or the ITSC department immediately in case of technical problems or service interuption in Moodle or related services.

|  |  |
| --- | --- |
| Student Name |  |
| Student ID No. |  |
| Class (L01 or L02) |  |

**PART A: Answer in this question booklet**

**Question 1 [4 marks]**

Study the following figures, which show part of the API documentation of the class ArrayList, and answer the questions on the next page. The contents are reproduced from [this link](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/ArrayList.html).

Graphical user interface, text

Description automatically generated

A

Graphical user interface, text, application, email

Description automatically generatedTeams

Description automatically generated with medium confidenceGraphical user interface

Description automatically generated

B

1. Which class is the super class / parent class of *ArrayList*? [1]
2. Suppose that a programmer wants to use an ArrayList by coding

ArrayList x = new ArrayList();

However, he finds that the code cannot run because he has forgot to include an import statement. What is the import statement that he should add to his code? [1]

1. Consider the methods equals and hashCode (see part B of the figure). Does an ArrayList object has these two methods? Where are they actually defined? [2]

**Practical tips for Written Questions 2 and 3: it is not easy to write codes using MS Words. If you wish, you may consider to write your code for this part using other software, and copy the answer here after you have finished.**

**Question 2 (6 marks)**

Write a Java method ***ticketPrice()***, which takes accept one int argument that represents the *number of tickets* that a customer is buying, and return the total ticket price, which is also an int.

The basic ticket price is for buying one ticket is $10.

However, if the customer buy more than 3 tickets in total, the price is reduced to $9 per ticket.

However, if the customer buy more than 6 tickets in total, the price is reduced to $8 per ticket.

For examples,

CallingticketPrice(1) should return 10*.*

CallingticketPrice(3) should return 30.

CallingticketPrice(4) should return 36.

CallingticketPrice(6) should return 54.

CallingticketPrice(7) should return 56.

You may assume the input is always non-negative. (note: you need to return the price rather than printing it).

**Note**: you do not need to write a complete program. You only need to write the ticketPrice method.

**Question 3 (10 marks)**

Given the following Staff class definition:

public class Staff {

public String name;

public int salary;

public char gender;

}

where gender is either 'M' or 'F'.

Supposed that we also have the following array:

Staff[] data = new Staff[100];

//and then some other codes to fill in the array content.

Assuming that the data[]array has been filled with an unknown number of Staff objects. Write some codes to find out and display the total salary of all male (gender 'M') staff. Note that some of the positions of the arrays can be empty.

**Note**: you do not need to write a complete program or a complete method. You only need to write the required code segments to compute the total.

**PART B: Long programming questions. Upload your works to Moodle**

**Question 4 (30 marks)**

**Note: put all programs in the default package.**

Write a Java Swing program (class name: StringFormatter), according to the following specification.

Graphical user interface, text, application

Description automatically generated

A **TextField** for user input

A **Button**

A **Button**

A **Label** for showing output

Graphical user interface, text, application

Description automatically generated**Requirement**

1. When the user press the **Upper Case** button, the program should convert the text entered on the TextField to upper case, and show the result in the label at the bottom. For example:

Graphical user interface, text, application

Description automatically generated

1. When the user press the **Lower Case** button, the program should convert the text entered on the TextField to lower case, and show the result in the label at the bottom. For example:

Marking Scheme: Reasonable GUI design (10%). Coding standard (10%). Functionality (80%)

**Question 5 (30 marks)**

**Note: put all programs in the default package.**

1. **Basic Part [28]**

Complete the command prompt program named NineNineNine, which is provided on **Moodle**.

* The program should keep reading integer inputs from the keyboard.
* For each input number that is not 999, you should simply store that number in the program (using a Stack).
* If the input is 999, you should print out all previously stored numbers in the reverse order, one

number per line. Then ends the program. (The number 999 itself does not need to be stored).

Here is an example execution of the program. User inputs are in blue.

>1

>2

>3

>4

>5

>6

>999

6

5

4

3

3

1

Good Bye!

**Note:**

* + **You must use a Stack to save the numbers**. Do not use other data structures.

**(Hints**: how to store the number if its not 999? What to do if it is 999? How to retrieve all numbers in the opposite order?).

* + A prompt symbol (>) should be used to indicate that the program is waiting for the next input.
  + You can assume all inputs are integer in the basic part.

1. **Advanced Part (Only for those who finish the rest of the exam early) [2]**

Handle invalid inputs by the users (such as non-integers) by simply ignoring them and continue to wait for the next input.

**End of Paper**