

ATHLONE INSTITUTE OF TECHNOLOGY

SCHOOL OF ENGINEERING

SEMESTER 1 EXAMINATIONS 2014

December Session



BACHELOR OF SCIENCE IN SOFTWARE DESIGN (WEB/GAME DEVELOPMENT)

YEAR 3

SOFTWARE DEVELOPMENT

External Examiner(s): Dr. Chris Exton
Mr. Damien Marshall

Internal Examiner(s): Dr. Enda Fallon

Instructions to candidates:

Read all questions carefully.

All questions carry equal marks.

Answer **Three** out of **Four** questions.

Time Allowed:

No. of pages including cover sheet: 5

- Q.1. (a) The following table stores details of student groups and grades. Assume that a class called Student **already exists** which encapsulates the data for student records including all necessary set and get methods.

Write a Main class which uses a collection such as an ArrayList to create and store the objects below.

(5 Marks)

Once the necessary objects have been created:

- Calculate the total number of students in all class groups
- Calculate the average student grade for the Game class group
- Calculate the average student grade for the Web class group
- Search and print out the ID of any student in the Web class group with a fail average (an average less than 40%)
- Student with ID=95 has recently moved from the Game to Web group. Outline the code to implement this change.

(7 Marks)

Student ID	Class Group	Average Grade %
12	Web	45
43	Game	78
78	Game	36
23	Web	76
7	Game	45
25	Web	67
37	Web	23
50	Web	65
95	Game	76
15	Game	62

- (b) Outline the updates required to the Student and Main classes in part (a) in order to

1. Serialize the objects to a file called "students.ser"
2. Deserialize the contents of "students.ser"

(8 Marks)

[20 Marks]

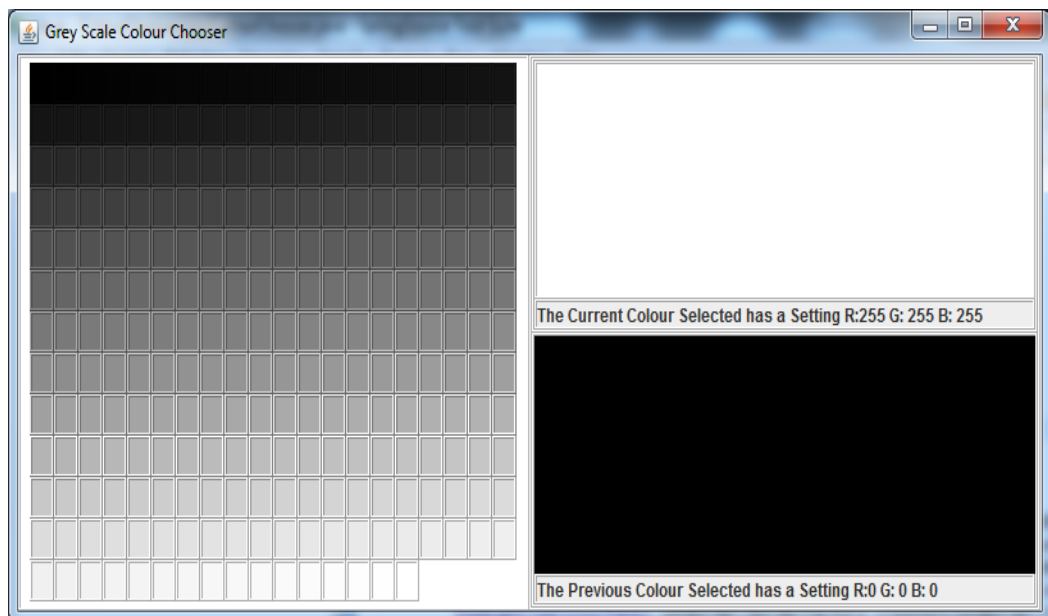
Q.2. (a) “Java is a platform independent language, however the Java Virtual Machine (JVM) is a platform dependent program.” Explain this statement using a diagram to illustrate the process by which Java source code is converted in to machine code for a particular operating system.

(7 Marks)

(b) A colour contains constituent components R (Red), G (Green) and B (Blue). In Java, a colour can be represented by these RGB components each of which is in the range 0-255. Black is represented by R=0, G=0 and B=0. White is represented by R=255, G =255 and B =255.

The following Graphical User Interface (GUI) represents the grey shades achieved by setting the RGB components to the equal values from R=0 G=0 B=0 (black), R=1G=1 B=1 R=255 G=255 B=255 (white). On the left of the main panel, buttons represent the colours. On the right of the main panel, panels and labels are used to indicate the current and previous colours selected.

Using sections of skeleton code outline how to implement the functionality below. In particular, your solution should (a) use a collection such as an ArrayList to store relevant GUI components (b) outline how event handling functionality is achieved (c) illustrate how colours are handled in a generic manner.



(13 Marks)

[20 Marks]

- Q.3. (a) “An Abstract class is a conceptual class which cannot be instantiated.” Explain this statement using a diagram and code examples. In your explanation describe (a) the purpose of abstract classes (b) why an abstract class can’t be instantiated (c) why an abstract class would be used in preference to an interface. (6 Marks)
- (b) Briefly explain the term “Design pattern”. In your explanation include a description of what a design pattern is, how design patterns evolved and name the elements of a design pattern. (5 Marks)
- (c) Explain using a diagram and short sections of code the Model View Controller (MVC) design pattern. Explain clearly the interaction between all components of the design pattern. (9 Marks)
[20 Marks]

- Q.4. (a) Explain using a diagram the components of a Java JSP Web application. Outline the functionality of each component. (5 marks)
- (b) Explain **using short sections of code** the functionality of the session object within a Java JSP application. (7 marks)
- (c) Using Java RMI , re-factor the following application so that we can access the Customer objects remotely.

```
Class Customer{
    private String theName;
    private String theAddress;

    public Customer (String aName, String aAddress)
    {
        theName = aName;
        theAddress = aAddress
    }
    public String getName()
    {
        return theName;
    }
    public int getAddress()
    {
        return theAddress;
    }
}
```

```
public class CustomerServer
{
    public static void main(String [] args)
    {
        Customer Tom = new Customer("Tom Hynes", "11 High Road");
        Customer Ann = new Customer("Ann Jones", "10 Low Road");
        System.out.println( "Customer Name: " + Tom.getName() + " lives at " +
                                Tom.getAddress());
    }
}
```

(8 Marks)
[20 Marks]