Athlone Institute of Technology

School of Engineering

Semester 1 Examinations 2015

December Session

Bachelor of Science in Software Design (Game/Cloud Development)

Year 3

Software Development

External Examiner(s): Dr. Chris Exton
Mr. Jerh O Connor

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: 2 Hours

No. of pages including cover sheet: 5

Q.1. (a) The following table stores details of mortgage customers including account number, mortgage type, outstanding mortgage balance and the number of months the mortgage is in arrears. Assume that a class called MortgageHolder **already exists** which encapsulates the data for mortgage 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.

(6 Marks)

Once the necessary objects have been created:

- Calculate the average mortgage balance for all customers
- Calculate the total number of customers in arrears
- Calculate the average number of months customers are in arrears
- Search and print the account numbers of any mortgage customer who is arrears for greater than 3 months.
- Calculate the average number of months customers with a tracker mortgage are in arrears
- Search and print the account numbers of any customer who has an outstanding mortgage balance in excess of €200,000 and is in arrears.

(6 Marks)

Mortgage Account	Mortgage Type	Balance €	Months in Arrears
7654	Variable	--63872	0
5473	Tracker	-231455	6
7383	Variable	-51562	2
7738	Tracker	-215278	0
9922	Fixed	-24155	0
8362	Fixed	-314555	7
7382	Variable	-132415	2

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

1. Serialize the objects to a file called “mortgage.ser”
2. Deserialize the contents of “mortgage.ser”

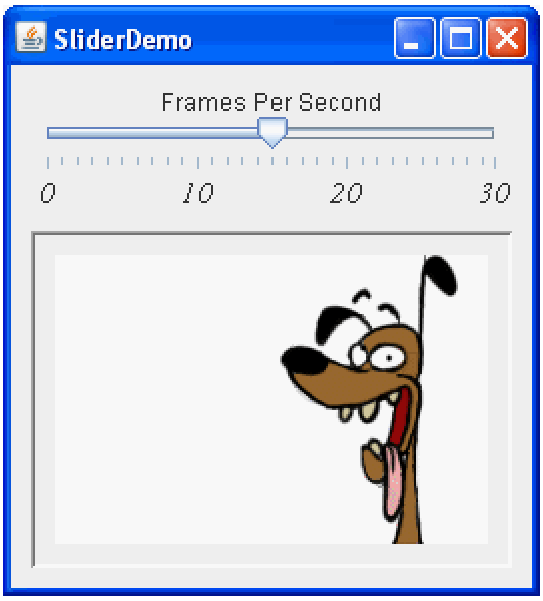
(8 Marks)

[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.

(8 Marks)

(b) Consider the following Graphical User Interface (GUI) which displays an animation of a dog. The GUI allows the end user to specify the number of Frames Per Second (FPS) which will be played.



Assume there are 30 files required to animate and these files are named dog1.gif, dog2.gif etc. Using sections of skeleton code outline how to implement the functionality of this GUI.

(12 Marks)
[20 Marks]

Q.3. (a) Explain the term “Java collections framework”. In your explanation include (a) the motivation for the framework (b) the 3 elements of the java collections framework (c) the 2 class hierarchies within the framework.

(7 Marks)

(b) With reference to the Java collections framework, explain using relevant sections of code the term iterator. In your explanation include (a) the purpose of an iterator (b) how to access an iterator (c) types of iterator.

(7 Marks)

(c) Outline using short sections of relevant code the operation of generics in the Java programming language.

(6 Marks)
[20 Marks]

Q.4. (a) Briefly explain the term “Design pattern”. In your explanation include a description of (a) what a design pattern is (b) how design patterns evolved (c) name the elements of a design pattern.

(5 Marks)

(b) “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)

(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());
   }
}
```

(9 Marks)
[20 Marks]