CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Semester 2 Examinations 2008/09

Module Title: Object-Oriented Programming 2

Module Code: **COMP7013**

School: **Computing**

Programme Title: **Bachelor of Science in Computing – Year 3**
Bachelor of Science (Honours) in Software Development – Year 2
Bachelor of Science (Honours) in Software Development & Computer Networking – Year 2

Programme Code: **KCOMP_7_Y3**
KSDEV_8_Y2
KDNET_8_Y2

External Examiner(s): **Ms. M. Meagher**
Internal Examiner(s): **Ms. D. M. Dunlea**
Mr. Rob Miller

Instructions: **Section A:** Choose 1 question
Section B: Choose 1 question
Section C: Choose 1 question

Duration: 2 Hours

Sitting: Summer 2009

Requirements for this examination:

Note to Candidates: Please check the Programme Title and the Module Title to ensure that you have received the correct examination paper.
If in doubt please contact an Invigilator.

Section A

Question 1

a) Create a BankAccount class that holds an ID, name and balance for a user. An interest of 4.5% is associated with all bank account classes. The class allows the user to retrieve the name and balance in the account. The ID number can be returned. The user can deposit or withdraw money from the account. The account may be created with a 0 balance or a specified balance.

[10 marks]

b) What is a class? What is an interface? What is the difference between a class and an interface in Java? [10 marks]

c) What modification would you make to the BankAccount class in order to implement the Measurable interface on it? The measure of a BankAccount can be considered to be its balance.

[8 marks]

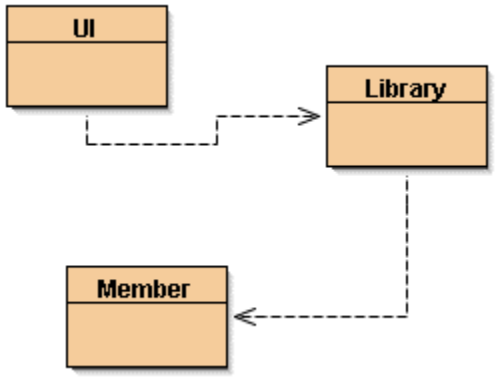
```
public interface Measurable
{
    double getMeasure();
}
```

d) Using the basic BankAccount class fulfil the following needs:
A class called PersonalDetails giving information about the owner of a BankAccount must be accessible. These must include address and phone no and PRSI number. A BankAccount cannot exist unless these details are provided. The BankAccount also keeps track of the last transactions on that account. A Transaction class should be created to deal with this. A transacton may either be a withdrawal, a deposit or interestAdded at the end of the month. Create these new classes and indicate any changes required to the original BankAccount class.

[12 marks]

Question 2

The following classes belong to an application that keeps track of items that are lent to members of a library:



The Classes are partially defined as follows:

```
public class UI extends JFrame implement ActionListener{

    …
    private  aLibrary = new Library();

    public UI(){}
    public void actionPerformed(ActionEvent e){}
    …
}
```

```
public class Library{

    private ArrayList<Membership> members;

    // The constructor opens a file and reads in member info
    public Library(){}

    // assume when members join they pay their subscription
    public void join (String name){}
    public int numberOfMembers(){}
    public deleteMember(String name){}

    // returns names of members whose subscriptions are due
    public String[] membersWithItemsDue(){}

    public void borrowItem(String itemName,String borrower){}
    public void returnItem(String name){}

    public void saveDataToFile(){}

}
```

```
public class Member{
        private String name;
        private Date returnDate;
        private String item;

        public Member(String){}
        public boolean itemDue(){}
        public void itemReturned(){}
        public String getName(){}
        public String getItem(){}

        // return false if an item already on loan to this
        //borrower

        public boolean itemBorrowed(String itemName){}
}
```

Points to Note:
Members can only borrow one item at a time;
Members can only borrow items for two weeks after that the items become due.

You are required to:

a) Provide a full implementation of the Library and Member classes.

[40 marks]

Section B

Question 3

a) How does inheritance promote software reusability? [8 marks]

b) What is an exception? A program can be designed to process an exception in one of three ways. What are these? Identify and explain the key mechanism used by Java when dealing with exceptions? What is the difference between a Checked and UncheckedException? Give one example of each. [10 marks]

c) Modify the code in the **BankAccount** *withdraw*() method so that an illegalArgumentException occurs if the user attempts to withdraw more money than exists in the account. Show how you would call this method from a test program. You can refer to Question 1a or create a basic BankAccount Class. [12 marks]

Question 4

a) Take a look at the following incomplete class definition:

```
public class ObjectQueue<X>
{
        …

        public ObjectQueue(){
                …
        }

        public int size(){
                …
        }

        public void add(X obj){
                …
        }

        public X get(){
                …
        }

        public X peek(){
                …
        }

}
```

Provide a complete implementation of the class

[14 marks]

b) Write the code for a java program that generates a random sequence of digits, displays them to the user for a second or so, and then asks the user to reproduce the sequence. Use queues to store the sequence of digits and the user's responses. The user should be given their results

[16 marks]

Section C

Question 5

a) A mySQL database exists on a server called mc-Admin. Write a program to show that a user name "dbadmin" can connect with a password of "students". What might cause your program to throw an exception? [10 marks]

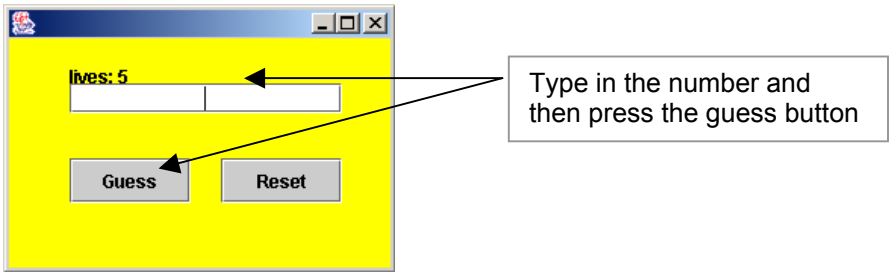
b) What is a thread? What steps are followed when running a simple thread in Java? [8 marks]

c) Create a greeting class which prints a greeting message every 30 seconds for 5 minutes and can run in its own thread. A message is passed to the greeting object when it is created. Write the code to test this class. [12 marks]

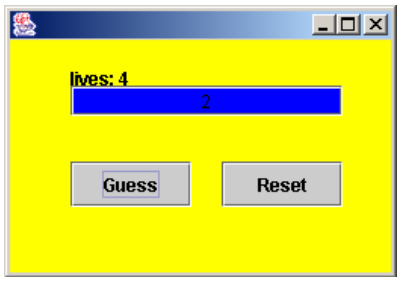
Question 6

Write the code for the simple game described below:

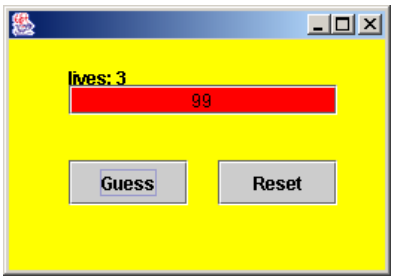
- Players must try and guess a secret number within a specified range e.g. 1 – 100.
- Players are allowed five attempts at guessing the number:



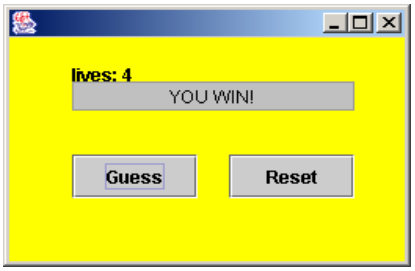
- If the guess is lower than the secret number, the background colour of the text-field turns blue:

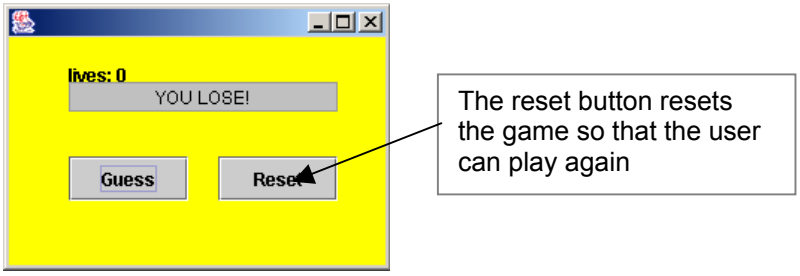


- If the guess is higher than the secret number, the background colour of the text-field turns red:



- The game finishes with either the player guessing the correct number or the player losing all their lives:





[30 marks]