CORK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA CHORCAÍ

Autumn 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 & 3

Programme Code: **KCOMP_7_Y3**
KSDEV_8_Y2
KDNET_8_Y2
KDNET_8_Y3

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: Autumn 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) Explain the term *superclass* and *subclass*. [8 marks]

b) Write a class for IDCard that inherits from the card class below. An IDCard is a card that holds a name and an ID number. The ID number of an IDCard is guaranteed to be unique. It should be possible to get the name and identifier of a card. To achieve this you should implement a method on IDCard that returns a string showing the name and id of the card as follows:
Cardholder: Susan
ID: 12

```
public class Card
{
        private String name;

        public Card()
        {
                name = "";
        }
        public Card(String n)
        {
                name = n;
        }
        public String getName()
        {
                return name;
        }
        public String toString()
        {
                return "Cardholder: " + name;
        }
}
```

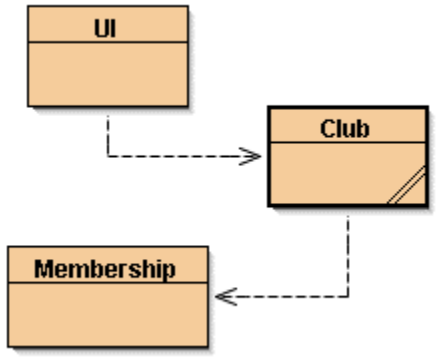
[10 marks]

c) What is meant by the term Polymorphism? Write a program that creates an ArrayList that contains both types of card and then iterates over the list to output the toString() method.
[10 marks]

d) What is generic programming?
Write a generic class called Pair that stores a pair of references of any type. The class should have two accessors getFirst and getSecond to access each of the references. [12 marks]

Question 2

The following classes belong to an application that keeps track of members in a Club:



The Classes are partially defined as follows:

```
public class UI extends JFrame implement ActionListener{

    …
    private Club aClub = new Club();

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

```
public class Club{

    private ArrayList<Membership> members;

    // The constructor opens a file and reads in membership info
    public Club(){}

    public void join (String name){}
    public int numberOfMembers(){}
    public deleteMember(String name){}

    // returns names of members whose subscriptions are due
    public String[] membersSubDue(){}
    public void paySub(String name){}
    public void saveDataToFile(){}

}
```

```
public class Membership{
        private String name;
        private Date subPaidDate;

        public Membership(String name){}
        public boolean isSubDue(){}
        public void subPaid(){}
        public String getName();
```

Points to Note:
When members join the club they pay their subscription.
A subscription lasts for one year after that it falls due.

You are required to:

a) Provide a full implementation of the Club and Membership classes.

[40 marks]

Section B

Question 3

a) Show the code required to allow the user to enter a string and store that string in a text file. [10 marks]

b) Describe how you would write an object to a file. [10 marks]

c) Show the code required to write an object to a file. [10 marks]

Question 4

Take a look at the following incomplete class definition:

```
public class ObjectStack<X>
{
        …

        public ObjectStack(){
                …
        }

        public int size(){
                …
        }

        public void push(X obj){
                …
        }

        public X pop(){
                …
        }

        public X peek(){
                …
        }

}
```

a) Provide a complete implementation of the ObjectStack class

[14 marks]

b) Write the code for a program that uses a stack to evaluate if a given string has a balanced set of braces i.e. the braces in the string "**abc{defg{ijk}{l{mn}}op}qr"** are balanced, while the braces in the string "**abc{def}}{ghij{kl}m"** are not balanced.

[16 marks]

Section C

Question 5

a) Why can deadlock occur when using locks? Explain a mechanism that may be used in Java to avoid deadlock. [8 marks]

b) If an instance of the BankAccount class (as covered in class) is accessed by more than one thread the balance variable must be protected by a lock. Show the changes required to the BankAccount class in order to synchronize access to this variable using locks [10 marks]

c) How would you define an enumerated type in Java? Why are they used? [4 marks]

d) Explain the following connect method which is part of the DBManager class. What errors may occur when executing the code? [8 marks]

```
class DBManager
{
        Connection dbConnection;

        public DBManager()
        {
                dbConnection = null;
        }

        public void connect()
        {
                String username = “dbadmin”;
                String password = “test”;
                String url = “jdbc:mysql://mc-Admin/users”;
                Class.forName(“com.mysql.jdbc.Driver”);
                dbConnection = DriverManager.getConnection(url, username, password);
        }
}
```

Question 6

a) What is a Layout manager? Describe the characteristics of the following layout managers:

- BorderLayout
- GridLayout
- FlowLayout
- BoxLayout

Are there any problems associated with the manual positioning and sizing of components?

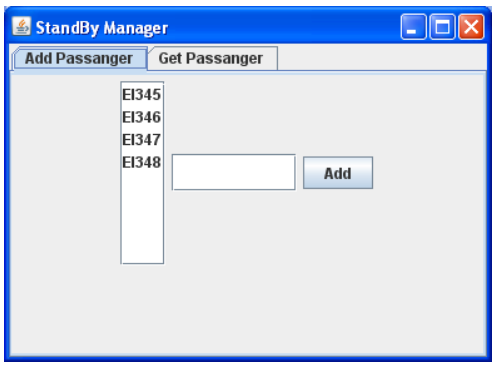
[12 marks]

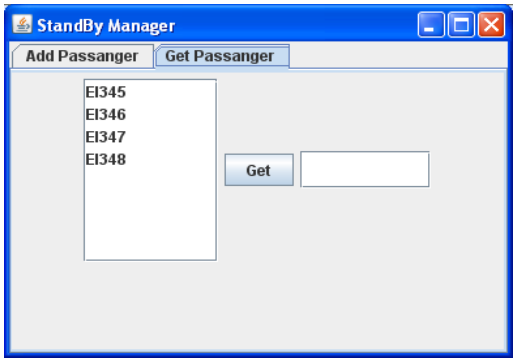
b) What do you understand by the term event-driven?

Explain what an inner class is. Why are they useful when writing event-handlers?

[6 marks]

c) Provide the code for the constructor of the following application (the constructor is responsible for setting-up the user interface):





[12 marks]