INSTITUTE OF TECHNOLOGY
TALLAGHT

Higher Certificate in Science
Bachelor of Science
Bachelor of Science (Honours)

Computing

Full Time

Semester Three : January 2016

Software Development 3

Internal Examiners
Ms. Patricia Magee

External Examiners
Dr. John Keating

Day	**Wednesday**
Date	**13th January 2016**
Time	**15:30-17:30**

Instructions to Candidates

Answer Question One and any two other questions

Question 1 **(40 Marks)**

Take a look at the class definitions below and answer the questions that follow:

```
package q1jan;

public abstract class Employee {

    private String name;

    public Employee(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return name;
    }

    public abstract double pay();
}

package q1jan;

public class FullTime extends Employee {

    private double salary;
    private double bonusRate;

    public FullTime(double salary, String name, double
    bonusRate) {
        super(name);
        this.salary = salary;
        this.bonusRate = bonusRate;
    }

    @Override
    public String toString() {
        return super.toString() + "\tSalaried\t";
    }

    @Override
    public double pay() {
        return salary;
    }

    public double calcBonus() {
        return salary * bonusRate;
    }
}
```

```
package q1jan;

public class PartTime extends Employee {

    private double rate;
    private double hours;
    private final int OVER_TIME = 40;

    public PartTime(double rate, double hours, String name) {
        super(name);
        this.rate = rate;
        this.hours = hours;
    }

    @Override
    public String toString() {
        return super.toString() + "\t Hourly\t";
    }

    @Override
    public double pay() {
        if (hours <= OVER_TIME) {
            return rate * hours;
        } else {
            return rate * OVER_TIME;
        }
    }

    public double calcOvertime() {
        double overtime = 0.0;
        if (hours > OVER_TIME) {
            overtime = (hours - OVER_TIME) * (rate * 1.5);
        }
        return overtime;
    }
}
```

a) Write the code for the test class to do the following using the sample output below as a guide:

i) Create a collection of `Employee` objects and store them in an `Arraylist`

(8 Marks)

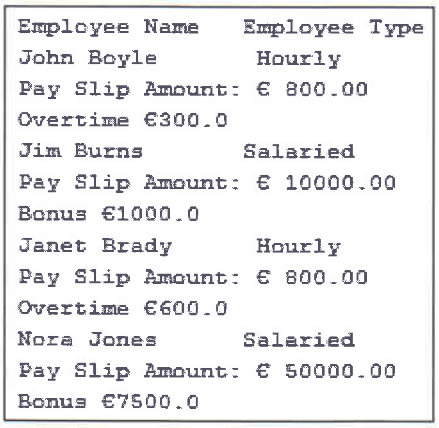
ii) loop through the `Arraylist` and display the employee information along with their pay slip amounts

(6 Marks)

iii) The FullTime class includes a `calculateBonus` method and the PartTime class includes a `calculateOverTime` method. You need to write the appropriate code to call these methods inside the loop you wrote in part ii). Explain the code that you have written.

(10 Marks)

Sample Output



```
Employee Name   Employee Type
John Boyle        Hourly
Pay Slip Amount: € 800.00
Overtime €300.0
Jim Burns        Salaried
Pay Slip Amount: € 10000.00
Bonus €1000.0
Janet Brady       Hourly
Pay Slip Amount: € 800.00
Overtime €600.0
Nora Jones       Salaried
Pay Slip Amount: € 50000.00
Bonus €7500.0
```

b) We now wish to keep track of the number of employees in the company. Explain with the aid of Java code how you would achieve this in the Employee class and the test class.

(6 Marks)

c) Write Java code that will sort the collection of employees by employee name in ascending order. Explain briefly the code you write to achieve this.

(10 Marks)

Question 2 **(30 Marks)**

The code below is a test class written to test a BankAccount class which has a private inner class called Transaction. The Transaction class is used to keep track of all the lodgements and withdrawals on a bank account. Answer the questions below to write the class definition for the BankAccount class.

```
package q2jan;

public class TestBankAccount {
    public static void main(String[] args) {
        BankAccount b = new BankAccount(777777,10000.0);
        b.deposit(5000);
        b.deposit(1000);
        b.deposit(2000);
        b.withdraw(6000);

        System.out.println(b);
    }
}
```

a) In the BankAccount class define the three members: account number (int), balance (double) and trnList (ArrayList of type Transaction). Write an appropriate constructor to initialise all the members. Use the call to the constructor in the test class as a guide.

(6 Marks)

b) Write a class definition for the Transaction class which has two members: transaction (String) and amount (double). Write an appropriate constructor which takes two parameters and write an appropriate toString() method.

(6 Marks)

c) Write a deposit method in the BankAccount class which updates the balance with the amount of the deposit and also creates a Transaction object and adds it to the ArrayList.

(5 Marks)

d) Write a withdraw method which updates the balance with the amount of the withdrawal and also creates a Transaction object and adds it to the ArrayList.

(5 Marks)

e) Explain with the aid of Java code two other types of inner classes in Java.

(8 Marks)

Question 3 **(30 Marks)**

a) Write a Java program that will write the 5 integer numbers 0,1,2,3,4 into a binary file. Your program will then read the contents of the file and find the sum of the numbers and display the sum on the screen.

(10 Marks)

b) Explain briefly the concept of streams in Java, outlining the two fundamental stream classes.

(8 Marks)

c) Consider the following code. The checkDay(String) method throws an InvalidDayException if the String given is not an actual day.

```
package q3bJan;

import java.util.Scanner;

public class PartB {

    public static void main(String[] args) {
        Scanner kb = new Scanner(System.in);
        String day;
        try {
            System.out.println("Enter a day?");
            day = kb.nextLine();
            checkDay(day);
            System.out.println("Thank you");
        } catch (InvalidDayException ice) {
            System.out.println("That’s not a day (:");
        } finally {
            System.out.println("Bye Bye ");
        }
    }
    public static void checkDay(String p) throws
    InvalidDayException {

        boolean found = false;
        String[] dayList = {"Monday", "Tuesday", "Wednesday",
               "Thursday", "Friday","Saturday", "Sunday"};

        for (String dayList1 : dayList) {
            if (dayList1.equals(p)) {
                found = true;
            }
        }
        if (found) {
            System.out.println("Valid");
        } else {
            throw new InvalidDayException();
        }
    }
}
```

i) On the first run of the program, the user enters "fffff" (which is not a day). Write the text that displays on the screen (input & output), in the order it is printed and typed.

(4 Marks)

ii) On another run of the program, the user enters "Monday" (which is a day). Write the text that displays on the screen (input & output), in the order it is printed and typed.

(4 Marks)

d) What is the difference between the keywords `throw` and `throws` in Java?

(4 Marks)

Question 4 **(30 Marks)**

A data set is a set of numeric values on which some calculations are to be performed. The following Java class defines a data set as an array of integer values:

```
package q4jan;

public abstract class DataSet {

    protected int[] data;

    // constructor
    DataSet(int[] data) {
        this.data = data;
    }
}
```

a) Write the Java code for a subclass of DataSet (i.e. MinDataSet) which will represent data sets whose minimum values are to be calculated inside a separate thread.

More specifically:

i) the constructor for the class should accept a reference to the array of integers that are to be used as the dataset

ii) when the thread is run it should calculate the minimum value in the data set and store the answer in a private member variable

iii) a method should be provided which returns the minimum value as calculated

(12 Marks)

b) The following code implements two classes: `TextMessage` and `TestMessage`. The `TextMessage` class represents a message that could be sent from a sender to a receiver. The `TestMessage` class creates a TextMessage object and passes this object as a parameter to two Receiver and one Sender constructor in classes that are not yet defined.

```
package q4bjan;

public class TextMessage {

    private String msg;

    public TextMessage(String str) {
        this.msg = str;
    }

    public String getMsg() {
        return msg;
    }

    public void setMsg(String str) {
        this.msg = str;
    }
}



package q4bjan;

public class TestMessage {
    public static void main(String[] args) {
        TextMessage msg = new TextMessage("Hello There");

        Receiver r1 = new Receiver(msg);
        Thread t1 = new Thread(r1,"Receiver 1");
        t1.start();

        Receiver r2 = new Receiver(msg);
        Thread t2 = new Thread(r2, "Receiver 2");
        t2.start();

        Sender s1 = new Sender(msg);
        Thread t3 = new Thread(s1, "Sender 1");
        t3.start();


        System.out.println("All the threads are started");
    }

}
```

Write the code for the `Receiver` class which has one member variable representing the text message and a constructor which initializes the member variable. Inside the `run` method, write code to display the name of the thread. Use a `synchronized` block to own the monitor of the `Receiver` text message. Display a message indicating that this thread is waiting and call on the appropriate method to wait for other threads to invoke notify methods. After the wait has ended, display a message "waiting thread got notified, message received"

(9 Marks)

c) Write the line numbers of the following programs which will generate compile errors and state the reasons why these errors arise:

(9 Marks)

```
1 package p1;
2
3 public class P1 {
4       public int p1_pub;
5       private int p1_pri;
6       protected int p1_pro;
7       int p1_def;
8 }


1 package p1;
2
3 public class P1SubP1 extends P1 {
4     public P1SubP1()
5    {
6         p1_pub = 5;
7         p1_pri = 5;
8         p1_pro = 5;
9         p1_def = 5;
10     }
11 }

1 package p1;
2
3 public class TestP1 {
4     public static void main(String[] args) {
5         P1 p1 = new P1();
6          p1.p1_pub = 5;
7          p1.p1_pro = 5;
8          p1.p1_def = 5;
9          p1.p1_pri = 5;
10     }
11 }
```

```
package p2;

import p1.P1;

public class P2 {
    public P2() {
          P1 p1 = new P1();
          p1.p1_pub = 5;
          p1.p1_pro = 5;
          p1.p1_def = 5;
          p1.p1_pri = 5;
      }
}
```

```
package p2;

import p1.P1;

public class P2SubP1 extends P1
{
    public P2SubP1() {
         p1_pub = 5;
         p1_pro = 5;
          p1_def = 5;
          p1_pri = 5;
      }
}
```

```
package p2;

public class P2SubP2 extends P2SubP1 {
    P2SubP2() {
         p1_pub = 5;
         p1_pro = 5;
         p1_def = 5;
         p1_pri = 5;
     }
}
```