Name:

_____

_____

Student Number:

_____

Please fill in your Student Number and Name.

Student Number : _____

University of Cape Town ~ Department of Computer Science

Computer Science 1015F ~ 2007

Supplementary Examination

Question	Max	Mark	Interna	External
1	15			
2	15			
3	15			
4	15			
5	4			
6	12			
7	12			
8	12			
TOTAL	**100**			

Marks : 100

Time : 3 hours

Instructions:

a) Answer all questions.

b) Write your answers in the space provided.

c) Show all calculations where applicable.

Question 1 [15]

Read the program below carefully.

```
import java.util.Scanner;
public class Question1supp
{
   public static void main ( String [] args )
   {
      for ( int i=0; i<3; i++ )
      {
         switch (i)
         {
            case 3:
               System.out.println ( "Me" );
               break;
            case 2:
               System.out.println ( "Fa" );
            case 1:
               System.out.println ( "Do" );
               break;
            default:
               System.out.println ( "Re" );
         }
      }
   }
}
```

a) Give an example of a selection statement from the program above. [1]

_____

b) Give an example of a constant used in the program above. [1]

_____

c) Give an example of a variable used in the program above. [1]

_____

d) Write down the exact output of this program. [4]

_____

_____

_____

e) Rewrite the **switch** statement using **if-else** statements. Just write down the replacement code. The new code should give exactly the same output as the original code. [5]

f) You now want to alter the program so that, instead of using the **for** statement, it will prompt the user to enter a number and will pass the number to the switch statement. Write down the code that you would add to do this. [3]

Question 2 [15]

Read the following Java program carefully.

```
public class Question2supp
{
   public static void main (String [] args )
   {
      double x = Math.random();
      int i = 0;
      while (i<3)
      {
         i++;
         if (x<0.5)
         {
            System.out.println ("Heads - you win!");
            break;
         }
         else
            System.out.println ("Tails");
         x = Math.random();
      }
      if ((x<0.5)&&(i==3))
         System.out.println("You won on the last throw!");

   }
}
```

a) Describe what this program does and what its outputs are. [3]

b) Give an example of an initialization statement in the program above. [1]

c) What is the difference between a **continue** statement and a **break** statement in a loop? [2]

d) Explain what happens if the **break** statement in the program is replaced with a **continue** statement. [1]

e) Rewrite the **main** method in this program so that it does not use the **break** statement and uses a **do-while** loop instead of a **while** loop. The program's output should be exactly the same as before. [5]

f) Explain how *lazy evaluation* of boolean expressions works, giving an example from the program listed above. [3]

Question 3 [15]

Consider the following program fragment:

```
public void doAction ()
{
   for ( int i=1; i<=x; i++ )
   {
      for ( int j=1; j<=(x-i); j++ )
         System.out.print (" ");
      for ( int j=1; j<=(2*i)-1; j++ )
         System.out.println ("*");
      System.out.println;
   }
}
```

a) What does this code do? [2]

b) **doAction** is a method. What is a method? [1]

c) What does the word **void** signify in the method header? [1]

d) The method uses a variable **x** that is neither defined locally nor passed in as a parameter. What type of variable is **x**? [1]

e) If we wish to test this method, we can use equivalence classes for input values.

i. What are equivalence classes? [2]

ii. Give an example of an equivalence class applicable to this program. [1]

f) Discuss 2 different ways to find errors in a program. [2]

g) This code is written in a high-level programming language. Discuss one major difference between a high-level and a low-level language. [2]

h) A modern computer that can execute such a program contains a CPU, memory (primary storage) and secondary storage (e.g., hard drives). Explain what the purpose of each of these parts is. [3]

Question 4 [15]

You need to calculate basic statistics for a class test, namely the average, highest and lowest marks. In the context of this problem, answer the following questions.

a) Describe an algorithm to calculate the statistics. Do not store all marks in an array. [4]

b) Describe how you would model your solution above using object-oriented programming (mention the major classes and methods). [3]

c) When we write classes, we usually adopt the principles of information hiding.

i. Why do we want information hiding? [1]

ii. What is the difference between the **private** and **public** modifiers? [2]

d) If some of your variables are declared as **private**, you need an *accessor* to access them from outside the class. Write an accessor method for one of the statistics. [2]

e) Some classes include methods that are prefixed with the word **static**.

i. What are **static** methods and variables? [1]

ii. Give one reason why we might possibly use them, and one reason why they should really be avoided. [2]

Question 5 [4]

You are given the following code fragment.

```
public class Data
{
   private int  pay;
   private double cost;
   public void what ( int val )
   {
      val = this.pay;
      System.out.println ("in what val = "+ val);
   }
   public void query ( Data dd )
   {
      dd.cost = this.cost;
   }
   public void set ( int i, double d )
   {
      pay = i;
      cost = d;
   }
}

public class Test
{
   public static void main ( String[] args )
   {
      Data place1 = new Data(),
           place2 = new Data();
      place1.set ( 5, 22.0);
      place2.set (10, 66.6);

      int val = 70;
      place2.what(val);
      System.out.println ("in Main val = "+ val);

      place1.query(place2);
      System.out.println ("in Main place1.cost="+place1.cost);
      System.out.println ("in Main place2.cost="+place2.cost);
   }
}
```

a) What is the output of the code fragment? [4]

Question 6 [12]

a) Write Java code to define a 1-dimensional array called **calc** that contains the following 7 values (2, -1, 3, -5, 8 , -8, 0), and write a loop that processes these values as follows:

- All zeroes have -10 added to them,
- All negatives are doubled and
- All positives are changed to negatives

E.g., (2, -1, 3, -5, 8, -8, 0) becomes (-2, -2, -3, -10, -8, -16, -10). [5]

Note: Do not put your code into a class or method – just write the Java statements.

b) Write Java code to print out each alternate value of the result from the previous question, for example (-2, -3, -8, -10). [1]

Note: Do not put your code into a class or method – just write the Java statements.

c) Write Java code to set the elements of an array to the values indicated below. Two loops must be sensibly used and no value may be read in from the keyboard. Define all variables used. The code should work for any square array. [6]

Note: Do not put your code into a class or method – just write the Java statements.

0	1	2	3	4
10	2	2	3	4
20	20	4	3	4
30	30	30	6	4
40	40	40	40	8

Question 7 [12]

Complete the two classes given below as instructed.

- A **Run** class with a **main** method that:
  - **Fill**s an array in the **Data** class
  - Finds the index of the row in the array that has the largest sum.
- A **Data** class that has a predefined method called **Fill**, as shown below, that assigns values to each element of the array.

a) Write a method called **RowWithLargestSum** that calculates the sum of the values in each row of the array and returns the index of the row with the largest sum. (A loop must be sensibly used). [10]

```
public class Data
{
   private int[][] value;
   private int[] calsum;

   public Data ( int maxRow, int maxCol )
   {
      // Constructor already written
      // An array with 6 rows x 5 columns is produced in
      // this example
   }

   public void Fill ()
   {
      // Already written method to fill entire array
   }

   // Write the RowWithLargestSum method
```

```
}
```

b) Complete the main method of the Run class. [2]

```
public class Run
{
   public static void main ( String [] args )
   {
      int maxRow = 6;
      int maxCol = 5;
      int largestRowIndex;
      Data x = new Data (maxRow, maxCol);
```

```
      System.out.println ("Row with largest sum is row" +
                          largestRowIndex);
   }
}
```

Question 8 [12]

Study the following incomplete class definitions. Use the concepts of *inheritance* and *polymorphism* in your answers to the questions that follow.

```
class Woman
{
   private String name;
   private int age;

   //parameterized constructor
   public Woman ( String n, int yrs )
   {
      name = n;
      age = yrs;
   }

   public String toString ()
   {
      return ("Name "+ name + "Age "+ age);
   }
}

class Wife extends Woman
{
   private String husbandName;
   public Wife (…); //parameterized constructor
   public String toString (…);
}

class Mother extends Wife
{
   private int numberKids;
   public Mother (…); //parameterized constructor
   public String toString (…);  //Assume this is given
}
```

a) Write parameterized constructors for **Wife** and **Mother** (The constructor for **Woman** is included in the class definition of **Woman**).

```
// Wife Constructor
```
[3]

```
// Mother Constructor
```
[2]

b) Write the **toString** method for **Wife** (The **toString** method for the class **Woman** has been given in the definition of **Woman**). You may assume that the **toString** method for **Mother** has been provided. [2]

c) Complete the **main** method that creates at least one parameterized object of each class (**Woman**, **Wife** and **Mother**). Print out all the data of each of those objects from inside a loop (thus illustrating the general method of doing this). [5]

```
public static void main ( String [] args )
{

}
```