Name: . . . . . .

ID Number: . . . . . .

Signature: . . . . . .

COMP102: Test

3 May, 2004

Instructions

- Time allowed: **2 hours**.
- Answer **all** the questions.
- There are 100 marks in total.
- Write your answers in the boxes in this test paper and hand in all sheets.
- If you think some question is unclear, ask for clarification.
- This test will contribute 25% of your final grade.
- Numeric keypad calculators and non-electronic dictionaries are permitted.

Questions	**Marks**	
1. Understanding Java	[35]	
2. Programs with conditionals	[10]	
3. Programs with loops	[10]	
4. Programs with objects	[20]	
5. Programs with arrays	[25]	
	TOTAL:	

Question 1. Understanding Java programs [35 marks]

For each of the following programs **(a)-(g)**, show the output produced when the program is run.

(a) [5 marks]

```
public class Test1 {
  public static void main(String[] args) {
    int x, y, z;
    x = 25;
    y = 4;
    z = 10;
    System.out.println(x + y * z);
    System.out.println(x / z);
    System.out.println("This is " + (x + y + z));
    System.out.println("That is " + x + " + " + y + " + " + z);
    System.out.println(Math.max(Math.min(x,y),z));
  }
}
```

(b) [7 marks]

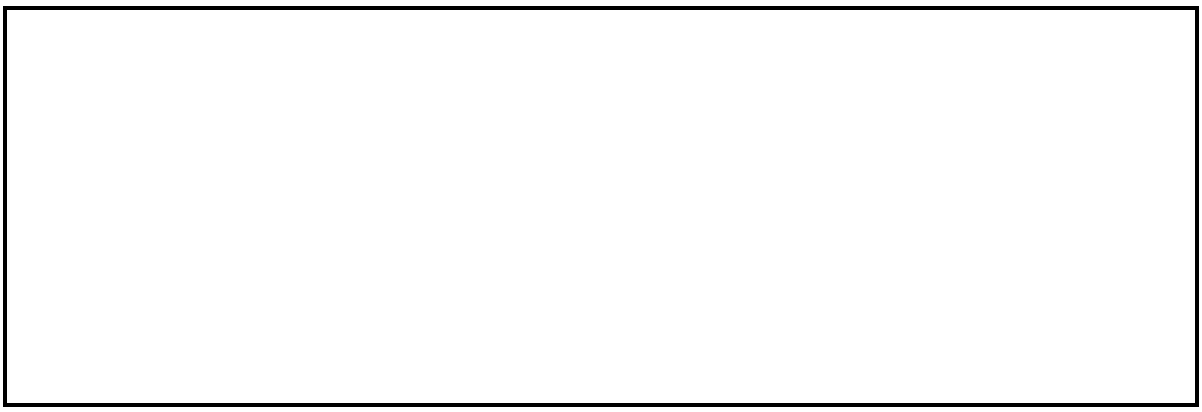
```
public class Test2 {
  public static void main(String[] args) {
    String s = "COMP 102";
    System.out.println(s.length());
    System.out.println(s.charAt(2));
    System.out.println(s.charAt(s.length()-1));
    int k = s.indexOf(" ");
    System.out.println(k);
    System.out.println(s.substring(0, k));
    System.out.println(s.substring(k+1));
    System.out.println("Next comes " + s.substring(0, k+1) +
      (Integer.parseInt(s.substring(k+1))+1);
  }
}
```

(c) [2 marks]

```
public class Test3 {
  public static void main(String[] args) {
    int a, b;
    a = 6;
    b = 15;
    if (b >= 2*a)
      System.out.print("red ");
    else
      System.out.print("blue ");
    System.out.println("grass");
  }
}
```

(d) [3 marks]

```
public class Test4 {
  public static void main(String[] args) {
    int a, b;
    a = 6;
    b = 15;
    c = 10;
    if (a < b && b < c)
      System.out.println("one");
    if (a < b && b > c)
      System.out.println("two");
    if (a > b || b < c)
      System.out.println("three");
    else
      System.out.println("four");
  }
}
```



(e) [3 marks]

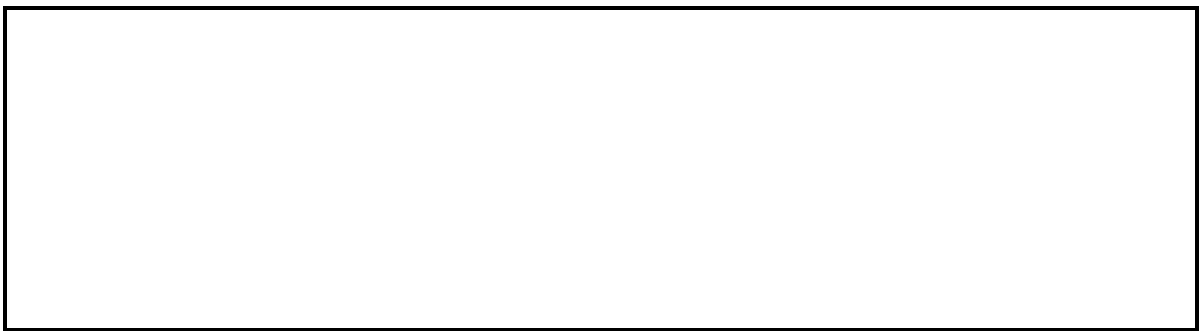
```
public class Test5 {
  public static void main(String[] args) {
    String s = "COMP 102";
    String t = "COMP 201";
    if ( s != null && t != null )
      if ( s.length() != t.length() )
        System.out.println("one");
      else
        if ( s.substring(0, 4).equals(t.substring(0, 4)) )
          System.out.println("two");
        else
          System.out.println("three");
    else
      System.out.println("four");
  }
}
```

(f) [6 marks]

```
public class Test6 {
  public static void main(String[] args) {
    int x, y, z;
    x = 1;
    y = 1;
    z = 0;
    while ( y < 12 ) {
      System.out.println(x + " " + y + " " + z);
      x = x+1;
      y = 2*y;
      z = z + y + 1;
    }
    System.out.println(x + " " + y + " " + z);
  }
}
```

(g) [9 marks]

```
public class Test7 {
  public static void main(String[] args) {
    int n = 2;
    for (int i = 0; i <= n; i++) {
      for (int j = n; j >= 0; j--) {
        int k = n*i+j;
        System.out.print(k + " ");
      }
      System.out.println();
    }
  }
}
```



Question 2. Programs with conditionals [10 marks]

Students sitting an exam receive a *Pass With Distinction* if they get a mark of 85 or more, a *Pass With Merit* if their mark is in the range 70 to 84, a plain *Pass* if their mark is in the range 50 to 69, and *Fail* if their mark is below 50.

Below is the outline of a program to read a student's name and mark, and print out the candidate's name and result. Complete the program, by adding the code required to set `res` to a string indicating the result for this student, as described above. You should assume that `s` is a string of digits, corresponding to a whole number between 0 and 100, inclusive.

```
import javax.swing.*;

public class ExamResults {

  public static void main(String args[]) {

    String name;
    int mark;
    String res;

    name = JOptionPane.showInputDialog("Enter name");

    // Read a mark and turn it into a number
    String s = JOptionPane.showInputDialog("Enter mark");
    mark = Integer.parseInt(s);

    // Set res to a string indicating the result for this student

    JOptionPane.showMessageDialog(null,
              "Result for " + name + " is " + res);
  }
}
```

Question 3. Programs with loops [10 marks]

Below is the outline of a program to print a multiplication table. The program is intended to read a positive integer, N, and then print a table with N rows and N columns, where each element contains the product of the row number and the column number (counting from 1 in each case). For example, if the input was 4, the table printed would be:

```
1 2 3 4
2 4 6 8
3 6 9 12
4 8 12 16
```

Complete the program, by adding the code required to print the table. You should assume that `s` is a string of digits, corresponding to a whole number, and print using `System.out.print` and/or `System.out.println`. (As in the example given, the columns are not required to line up.)

```
import javax.swing.*;

public class MultiplicationTable {

  public static void main (String[] args) {

    String s = JOptionPane.showInputDialog("Enter table size");
    int size = Integer.parseInt(s);

    // Print multiplication table

  }
}
```

SPARE PAGE FOR EXTRA ANSWERS

Cross out rough working that you do not want marked.
Specify the question number for work that you do want marked.

Question 4. Programs with objects [20 marks]

The following is a simple program for constructing a payroll:

```
public class PayMaster {
  public static void main(String[] args) {
    Payroll p = new Payroll();
    p.addItem("Chris", 4, 10.50);
    p.addItem("Mike", 2, 10.50);
    p.addItem("Mary", 3, 15.00);
    p.showPayroll();
  }
}

class Payroll {
  private String payroll = "";
  public Payroll() {
  }
  public void addItem(String name, int hours, double rate) {
    double pay = hours * rate;
    payroll = payroll + name + " " + hours + "@" + rate + " = " + pay +
    "\n";
  }
  public void showPayroll() {
    System.out.println(payroll);
  }
}
```

(a) [10 marks] What output will this program produce?

(b) [10 marks] Suppose you want a call on `showPayroll` to also print the total amount to be paid out. Show the changes you would need to make on the copy of the `Payroll` class given in the answer box below. (You should not change the `PayMaster` class.)

```
class Payroll {


  private String payroll = "";


  public Payroll() {


  }


  public void addItem(String name, int hours, double rate) {


    double pay = hours * rate;


    payroll = payroll + name + " " + hours + "@" + rate + " = " + pay +
    "\n";


  }


  public void showPayroll() {


    System.out.println(payroll);


  }


}
```

Question 5. Programs with arrays [25 marks]

(a) [9 marks] What output will the following program produce?

```
public class ArrayProg1 {
  public static void main(String[] args) {
    int[] nums = {10, 25, 15, 80, 4};
    System.out.println(nums.length);
    int i = 2;
    System.out.println(nums[i]);
    System.out.println(2*nums[i]);
    System.out.println(nums[nums[nums.length-1]-1]);
    for (int k = nums.length-1; k >= 0; k--)
      System.out.print(nums[k] + " ");
  }
}
```

(b) [8 marks] What output will the following program produce?

```
public class ArrayProg2 {
  public static void main(String[] args) {
    int[] nums = {1, 2, 3, 4, 5, 6, 7, 8};

    for (int i=0; i < nums.length; i++)
      nums[i] = nums[i]*10;

    for (int k = 0; k < nums.length; k++)
      System.out.print(nums[k] + " ");
    System.out.println();

    for (int i=0; i < nums.length/2; i++)
      nums[i] = nums[i]+nums[nums.length-i-1];

    for (int k = 0; k < nums.length; k++)
      System.out.print(nums[k] + " ");
    System.out.println();

  }
}
```

(c) [8 marks] Complete the following method, so that it takes two arguments, an integer `item` and an array of integers `nums`, and returns the number of times that `item` occurs in `nums`.

```
// Count the number of times that item occurs in nums
public int countOccurrences(int item, int[] nums) {
```

```

```

}

****