

Name:

ID Number:

Signature:

COMP102: Test

18 April, 2005

Instructions

- Time allowed: $1\frac{1}{2}$ 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. Simple classes and conditionals

[15]

3. Loops

[15]

4. Arrays

[18]

5. Arrays of objects

[17]

TOTAL:

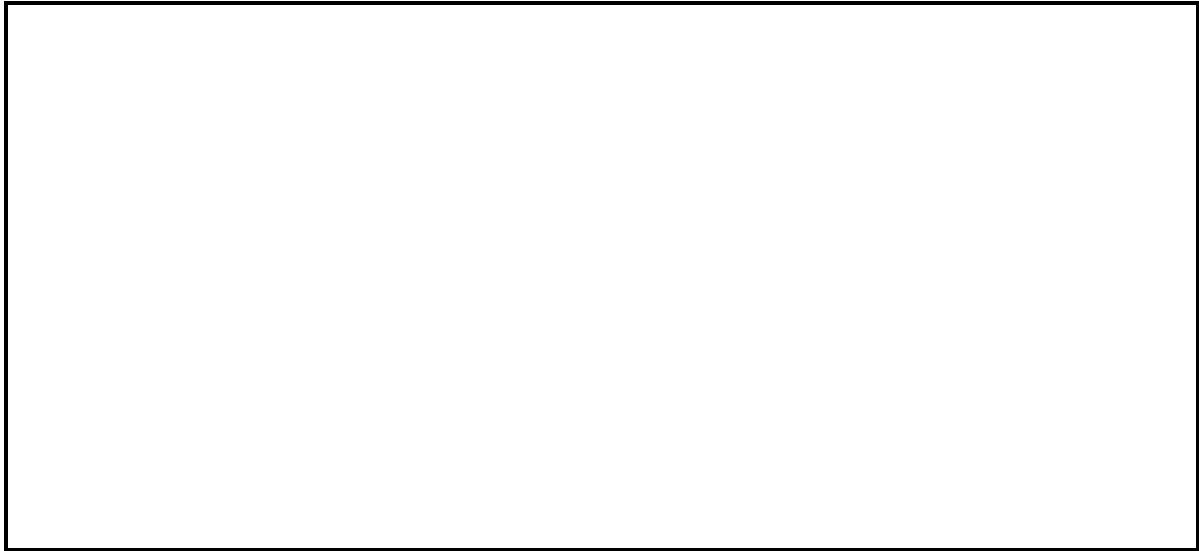
Question 1. Understanding Java

[35 marks]

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

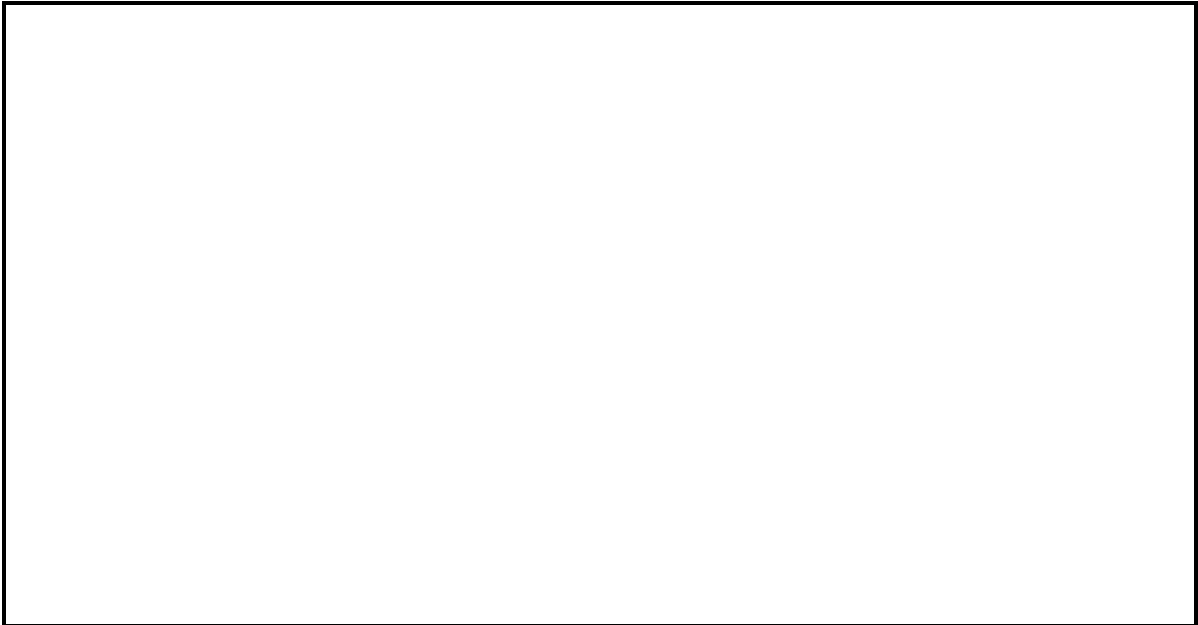
(a) [6 marks]

```
public class Test1 {  
    public static void main(String[] args) {  
        int x = 12;  
        int y = 10;  
        int z = 4;  
        System.out.println("1: " + x + " + " + y + " + " + z);  
        System.out.println("2: " + (x + y + z));  
        System.out.println("3: " + (x * z - y));  
        System.out.println("4: " + y / z);  
        System.out.println("5: " + y / 4.0);  
        System.out.println("6: " + Math.min(y, Math.max(x, z)));  
    }  
}
```



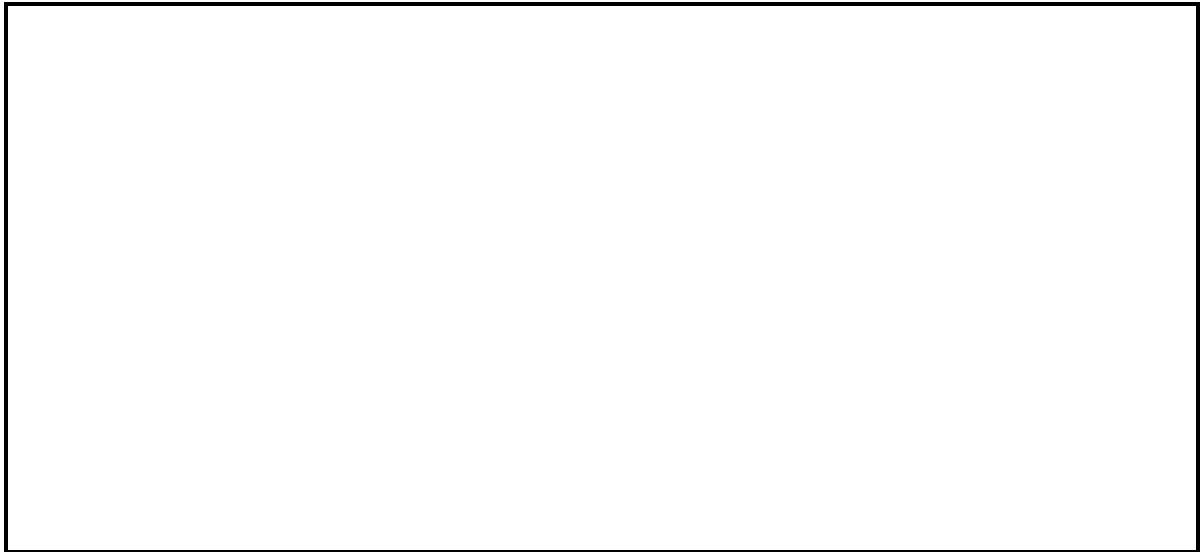
(b) [6 marks]

```
public class Test2 {  
    public static void main(String[] args) {  
        String a = "9876543210";  
        String b = "Computers are fun";  
        String c = "Say it again!";  
        int i = a.indexOf("1");  
        int j = b.indexOf(" ");  
        int k = c.indexOf(" ");  
        System.out.println(i);  
        System.out.println(j);  
        System.out.println(k);  
        System.out.print(c.charAt(4));  
        System.out.println(b.charAt(8));  
        System.out.print(b.substring(0, j+1));  
        System.out.println(b.substring(b.length()-3));  
    }  
}
```



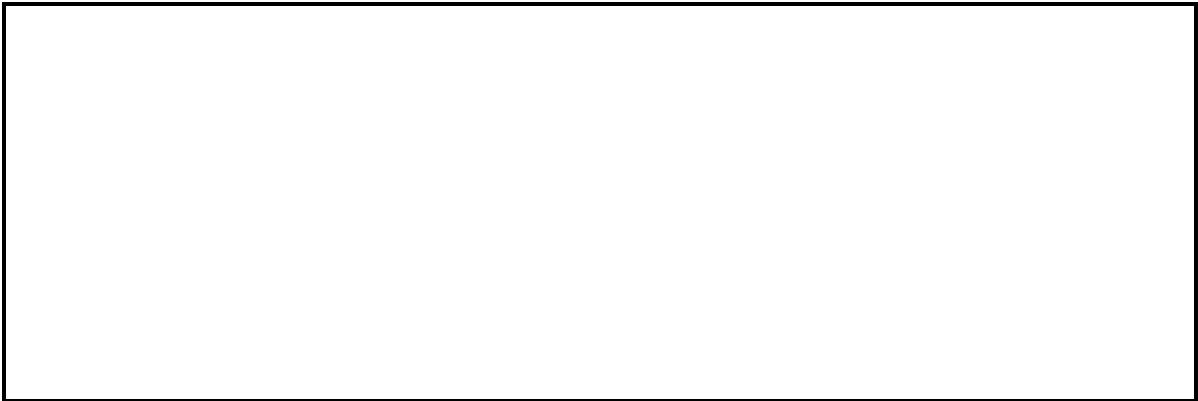
(c) [6 marks]

```
public class Test3 {
    public static void main(String[] args) {
        int a = 25;
        int b = 10;
        int c = 5;
        String r = null;
        String s = "";
        if ( a-c >= 2*b )
            r = "cat";
        else
            r = "dog";
        System.out.println(r);
        if ( b > c && c < a )
            s = "food";
        System.out.println(s);
        if ( r == null || s == null )
            System.out.println("Error!");
        else
            if ( r.length() < s.length() )
                System.out.println("Buy " + r + " " + s);
            else
                System.out.println("Feed the " + r);
    }
}
```



(d) [7 marks]

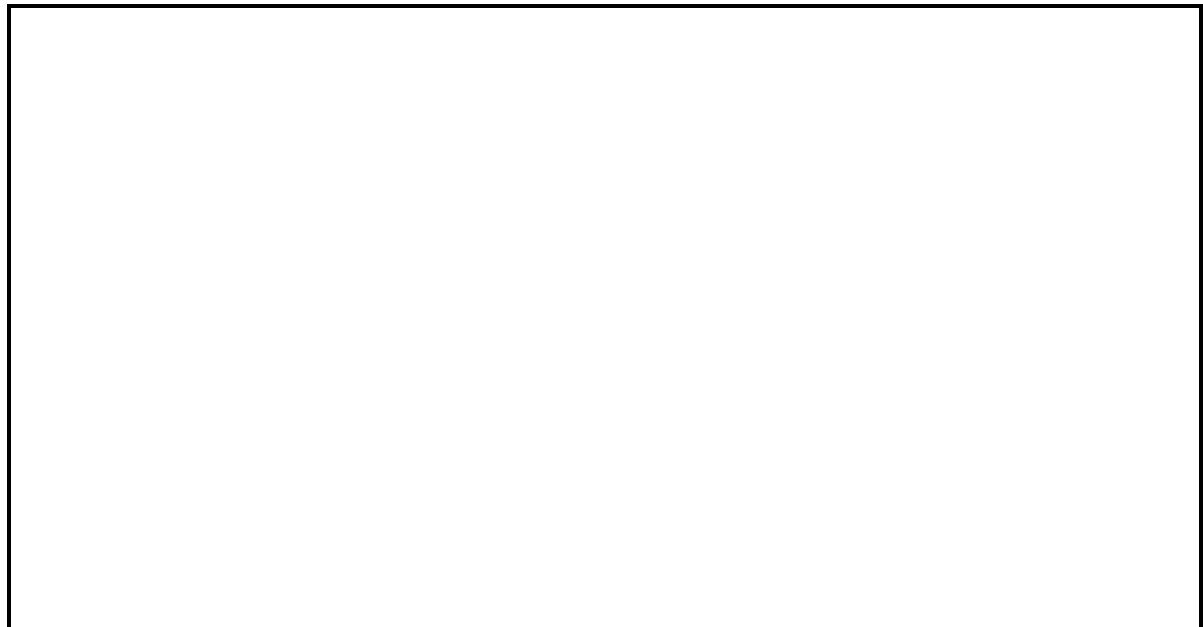
```
public class Test4 {  
    public static void main(String[] args) {  
        int x = 0;  
        for (int i = 1; i < 2; i++) {  
            int y = 0;  
            for (int j = i; j < 4; j++) {  
                y = y + j;  
                System.out.println("y = " + y);  
            }  
            x = x + y;  
            System.out.println("x = " + x);  
        }  
    }  
}
```



(e) [10 marks]

```
public class Test5 {
    public static void main(String[] args) {
        Order ord = new Order();
        ord.enter("Bread", 1, 2.20);
        ord.enter("Milk", 2, 3.40);
        ord.enter("Beans", 5, 1.20);
        ord.printFinal();
    }
}

class Order {
    private double sum;
    private double tot;
    public Order() {
        sum = 0;
        tot = 0;
    }
    public void enter(String s, int q, double p) {
        double r = q*p;
        sum = sum + q;
        tot = tot + r;
        System.out.println( s + " " + q + " " + r );
    }
    public void printFinal() {
        System.out.println( sum );
        System.out.println( tot );
    }
}
```



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.

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 2. Simple Classes and Conditionals

[15 marks]

A motel charges \$50 per person per night for up to two people, and \$10 per night for each additional person.

Below is the outline of a program to compute the cost of a room for a given number of people for a given number of nights.

Complete the program, by adding the code required for the RoomHire constructor, the getCost method, and any fields required in RoomHire.

Note that the program is not required to check the validity of the inputs read, and you should not change ComputeRoomCost.

```
import javax.swing.*;
public class ComputeRoomCost {
    public static void main(String[] args) {
        // Read the number of people
        String s = JOptionPane.showInputDialog("How many people?");
        int people = Integer.parseInt(s);
        // Read the number of nights
        s = JOptionPane.showInputDialog("How many nights?");
        int nights = Integer.parseInt(s);
        RoomHire r = new RoomHire(people, nights);
        System.out.println("Cost = " + r.getCost());
    }
}
```

```
class RoomHire {

    public RoomHire(int p, int n) {

    }

    public double getCost() {

    }

}
```

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 3. Loops

[15 marks]

Complete the following program so that it reads an integer n and prints n lines, each consisting of n characters. The first and last character on each line is an asterisk and all other characters are minuses. For example, if the input is 5, the program should print:

```
*---*
*---*
*---*
*---*
*---*
```

You should assume that the input entered is an integer greater than one.

```
import javax.swing.*;
public class PrintLines {
    public static void main(String args[]) {
        // Read the n as a string and turn it into an int
        // Assume that n is an integer greater than 1
        String s = JOptionPane.showInputDialog("Enter size");
        int n = Integer.parseInt(s);
        // Print n lines of asterisks and minuses
```

```
    }
}
```

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. Arrays

[18 marks]

Complete the following program so that it reads an integer n , creates an array of size n , and reads n integers and stores them in the array. It should then compute and print the sum of the integers in the array, and the number of zero values in the array.

You may assume that the value of n entered is a valid, non-negative integer, and that the other numbers read are valid integers.

```
public class ArrayProcessor {
    public static void main(String[] args) {
        // Read n
        String s = JOptionPane.showInputDialog("Enter n");
        int n = Integer.parseInt(s);

        // Create an array of size n.

        // Read n integers and store them in the array.

        // Compute and print the sum of the integers in the array,
        // and the number of zero values in the array.

    }
}
```

Question 5. Arrays of objects

[17 marks]

You are part of a team required to write a program that maintains a list of people and their birthdays. Your task is to implement a `PersonList` class, which stores the list of people and their birthdays. Other members of the team will write the main program, which interacts with the user and calls methods in your `PersonList` class to perform operations requested by the user.

The team leader has decided what methods the `PersonList` class will have, and has given you the following outline showing the method headers and describing their intended behaviour. The list is to be implemented as an array of `Person` objects, each storing information about one person. The code for the `Person` class is shown below.

Complete the definition of `PersonList` so that the methods behave as describe in the comments.

```
class Person {
    private String name;
    private String birthday;
    public Person(String n, String b) {
        name = n;
        birthday = b;
    }
    public String getName() {
        return name;
    }
    public String getBirthday() {
        return birthday;
    }
}
```

```
class PersonList {
    // Data fields

    // Constructor: create an empty list with the given maximum size.
    public PersonList(int MaxSize) {

    }

    // Add a person and his/her birthday to the list.
    // If the list is full, print a message and leave the list unchanged.
    // Assume that the person is not already in the list.
    public void enter(String name, String bday) {

    }

    // Find the birthday for a given person.
    // Return null if the person is not in the list.
    public String find(String name) {

    }
}
```

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.

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.