THE UNIVERSITY OF AUCKLAND

SUMMER SEMESTER, 2004

Campus: City

COMPUTER SCIENCE
TEST
Principles of Programming

(Time allowed: 75 MINUTES)

NOTE: Attempt **ALL** questions
Write your answers in the space provided
There is space at the back for answers that overflow the allotted space
No calculators are permitted

Surname:	
Forenames:	
Student ID number:	
Login name:	

Question	Marks	Out of
Question 1 (output)		21
Question 2 (explain errors)		9
Question 3 (boolean, if statement)		10
Question 4 (write a method)		10
Question 5 (write a method)		10
Question 6 (arrays)		10
Question 7 (GregorianCalendar)		8
Question 8 (arrays)		10
Question 9 (classes)		12
TOTAL		**100**

SURNAME: ..... FORENAMES: .....

Question 1 (21 marks)

a) What is printed by the following?

```
System.out.println(10 + 9 + "a");
```

(1 mark)

b) What is printed by the following?

```
System.out.println("5 + 1.5");
```

(1 mark)

c) What is printed by the following?

```
String s = new String("mississippi");
System.out.println(s.indexOf("ss"));
```

(1 mark)

d) What is printed by the following?

```
String ss = new String("basketball");
System.out.println(ss.substring(1,4) + " " + ss.substring(6));
```

(1 mark)

SURNAME: ..... FORENAMES: .....

e) What is printed by the following?

```
System.out.println(10 / 5.0);
```

(1 mark)

f) What is printed by the following?

```
double b = 11 / 5;
System.out.println(b);
```

(1 mark)

g) What is printed by the following?

```
String s1 = new String("a");
String s2 = new String("a");
String s3 = s2;
System.out.println(s3.equals(s1));
```

(1 mark)

h) What is printed by the following?

```
Point p = new Point(5, 6);
p.setLocation(10, 1);
System.out.println("[" + (int)p.getY() + ","
                                        + (int)p.getX() + "]");
```

(2 marks)

SURNAME: ..... FORENAMES: .....

i) What is printed by the following?

```
System.out.println(8 % 10);
```

(2 marks)

j) What is printed by the following?

```
System.out.println("\\n\"");
```

(2 marks)

k) What is printed by the following?

```
System.out.println(5 / 4 + (6 - 3 * 7) + 2 / 8 - 1 % 9);
```

(2 marks)

l) What is printed by the following?

```
System.out.println(Math.min(Math.max(5, 6), Math.min(5, 6)));
```

(2 marks)

SURNAME: ..... FORENAMES: .....

m) What is printed by the following?

```
System.out.println(Math.pow(2,3) + Math.sqrt(9));
```

(2 marks)

n) What is printed by the following?

```
System.out.println(Double.parseDouble(10 + "." + 15));
```

(2 marks)

SURNAME: ..... FORENAMES: .....

Question 2 (9 marks)

For each of the questions below, you need to provide an explanation.

a) Why does the following code **not** print out a random value between 1 and 10 (inclusive)?

```
int rand = ((int)Math.random())*10 + 1;
System.out.println(rand);
```

Please explain:

(3 marks)

b) Why does the following loop **not** print out the integers from 0 up to and including 4?

```
int i = 0;
while (i < 5)
     System.out.println(i);
     i++;
```

Please explain:

(3 marks)

c) Why does the following code **not** compile?

```
int num = Integer.parseInt(204);
```

Please explain:

(3 marks)

SURNAME: ..... FORENAMES: .....

Question 3 (10 marks)

(a) What is the output when the following application is executed?

```
public class Q3A {

     public static void main(String[] args) {

          int num1 = 15;
          int num2 = 20;
          int num3 = 12;
          boolean hasFinished = false;

          boolean result = (num1<num2 && !(num3>num1));
          System.out.println("1. " + result);

          result = !result;
          System.out.println("2. " + result);

          result = (num2<num1 || num3!=num3);
          System.out.println("3. " + result);

          if (!hasFinished)
               System.out.println("4.");
          else
               System.out.println("5.");
     }
}
```

```
> java Q3A
```

(4 marks)

CONTINUED

SURNAME: ..... FORENAMES: .....

(b) What is the output when the following application is executed?

```
public class Q3B {

     public static void main(String[] args) {

          System.out.println("1.");
          whatIf(5, 10);

          System.out.println("2.");
          whatIf(13, 6);

     }

     private static void whatIf(int x, int y) {

          if (x < 10) {
               if (y > 9 && y != 5) {
                    System.out.println("line 1");
                    x = x + 1;
               }
               if (y%2 == 0 || y == 3)
                    System.out.println("line 2");
               else
                    System.out.println("line 3");
          }
          else if (y <= 9) {
               if (y < x && y >= 5){
                    System.out.println("line 4");
                    x = 12;
               }
               System.out.println("line 6");
               x = x - 1;
          }
          if (x%2 == 0)
               System.out.println("line 7");

          System.out.println("line 8");
     }
}
```

SURNAME: ..... FORENAMES: .....

```
> java Q3B
```

(6 marks)

SURNAME: ..... FORENAMES: .....

Question 4 (10 marks)

Write a method `fillOutString()` which accepts two parameters. The first parameter is a String representing a word or phrase and the second parameter is an `int` representing the total number of characters the filled out String should contain. Star characters ('*') are added to the left of the String parameter until the whole String has a length equal to the required number of characters. The method returns the String after it has been filled out with the correct number of '*' characters. If the String passed in as a parameter is longer than the total number of characters, the method returns the String itself without any added padding.

Executing the `Q4` application with the completed `fillOutString()` method produces the following output:

```
> java Q4
123456789012345678901234567890123456
**GOOD LUCK!
*******GOOD!
*******WELL DONE
Summer School is Great
```

```
public class Q4 {

     public static void main(String[] args) {

          System.out.println("123456789012345678901234567890123456");

          String str = "GOOD LUCK!";
          str = fillOutString(str, 12);
          System.out.println(str);

          str = "GOOD!";
          str = fillOutString(str, 12);
          System.out.println(str);

          str = "WELL DONE";
          str = fillOutString(str, 16);
          System.out.println(str);

          str = "Summer School is Great";
          str = fillOutString(str, 8);
          System.out.println(str);
}
```

SURNAME: ..... FORENAMES: .....

```
 private static _______ fillOutString(_____________________________){

 }
}
```

(10 marks)

SURNAME: ..... FORENAMES: .....

Question 5 (10 marks)

Write a method, `printRating()` which accepts three `int` parameters. The first parameter is the number of A ratings, the second parameter is the number of B ratings and the third parameter is the number of C ratings. The method prints the average of all the ratings (the average is the total value of all the ratings divided by the total number of ratings). To work out the average, a value is given to each rating category:

each A rating is worth 10
each B rating is worth 6
each C rating is worth 3

As well as displaying the average, the method displays a certain number of stars:

five stars ("`*****`") if the average is greater than 8.5
three stars ("`***`") if the average is greater than 6 but not greater than 8.5
one star ("`*`") in all other cases.

Executing the `Q5` application with the completed `printRating()` method produces the following output:

```
> java Q5

8, 0, 2
Rating: 8.6 *****

1, 9, 0
Rating: 6.4 ***

0, 4, 7
Rating: 4.090909090909091 *
```

```
public class Q5 {

     public static void main(String[] args) {

          System.out.println();
          System.out.println("8, 0, 2");
          printRating(8, 0, 2);

          System.out.println();
          System.out.println("1, 9, 0");
          printRating(1, 9, 0);

          System.out.println();
          System.out.println("0, 4, 7");
          printRating(0, 4, 7);
     }
}
```

CONTINUED

SURNAME: ..... FORENAMES: .....

```
private static ___ printRating(_____________________________________){









  }
}
```

(10 marks)

SURNAME: ..... FORENAMES: .....

Question 6 (10 marks)

Write the method `lengthSum()` which accepts an array of `String`s as a parameter. The method should then calculate and return the total length of all of the `String`s in that array.

For example, executing the `Q6` application (given below) with the completed `lengthSum()` method produces the following output:

```
> java Q6
15
4
```

Carefully examine the source code for the application `Q6` below, and then complete the source code for the `lengthSum()` method on the following page:

```
public class Q6 {

     public static void main(String[] args) {

          String[] words1 = {"apple", "banana", "lime"};
          System.out.println( lengthSum(words1) );

          String[] words2 = {"a", "b", "cd"};
          System.out.println( lengthSum(words2) );

     }
}
```

SURNAME: ..... FORENAMES: .....

```
public static int lengthSum(String[] words) {
```

(10 marks)

```
    }
}
```

SURNAME: ..... FORENAMES: .....

Question 7 (8 marks)

Write a method, `getYearWhen()`, which accepts two parameters. The first parameter is a `GregorianCalendar` object representing a date of birth. The second parameter is an `int` representing an age greater than 0. The method returns the year when the age given by the second parameter is (or was) reached.

Executing the `Q7` application with the completed `getYearWhen()` method produces the following output:

```
> java Q7

You were/are/will be 50 in: 2004

You were/are/will be 21 in: 2022

You were/are/will be 20 in: 1990
```

```
import java.util.*;

public class Q7 {

     public static void main(String[] args) {

          GregorianCalendar gCal = new GregorianCalendar(1954,
                                                         4, 23);
          int wantedYear = getYearWhen(gCal, 50);
          System.out.println("You are/were/will be 50 in: "+
                                                    wantedYear);

          System.out.println();
          gCal = new GregorianCalendar(2001, 10, 5);
          wantedYear = getYearWhen(gCal, 21);
          System.out.println("You were/are/will be 21 in: "+
                                                    wantedYear);

          System.out.println();
          gCal = new GregorianCalendar(1970, 3, 15);
          wantedYear = getYearWhen(gCal, 20);
          System.out.println("You were/are/will be 20 in: "+
                                                    wantedYear);

     }
}
```

SURNAME: ..... FORENAMES: .....

```
private static ___ getYearWhen(_____________________________________){

}
}
```

(8 marks)

SURNAME: ..... FORENAMES: .....

Question 8 (10 marks)

Write the method `doubleArray()` which accepts an array of `int`s as a parameter, and returns a new array of `int`s where the value of each element in the new array is twice the value of the corresponding element in the original array.

For example, executing the `Q8` application (given below) with the completed `doubleArray()` method produces the following output:

```
> java Q8
2 4 6
```

Carefully examine the source code for the application `Q8` below, and then complete the source code for the `doubleArray()` method on the following page:

```
public class Q8 {

     public static void main(String[] args) {

          int[] a = {1, 2, 3};
          int[] b;

          b = doubleArray(a);

          System.out.println(b[0] + " " + b[1] + " " + b[2]);

     }
}
```

SURNAME: ..... FORENAMES: .....

```
    public static int[] doubleArray(int[] nums) {
```

(10 marks)

```
    }
}
```

SURNAME: ..... FORENAMES: .....

Question 9 (12 marks)

Examine the source code for the following application carefully:

```
public class Q9 {

     public static void main(String[] args) {

          Camera myCamera, yourCamera;

          myCamera = new Camera(true, 0);
          yourCamera = new Camera(false, 0);

          System.out.println(myCamera);
          System.out.println(yourCamera);

          myCamera.takePhotos(7);
          yourCamera.takePhotos(4);

          System.out.println(myCamera);
          System.out.println(yourCamera);

          myCamera.setOn(false);
          yourCamera.setOn(true);
          yourCamera.takePhotos(3);
          yourCamera.takePhotos(2);

          System.out.println(myCamera);
          System.out.println(yourCamera);
     }
}
```

The application `Q9` above uses a `Camera` class to create two `Camera` objects. When a `Camera` object is constructed, two values are specified to the constructor method. The first value is a `boolean` which indicates whether the camera is turned on or not, and the second value is an `int` which specifies the number of photos that the camera has taken so far (this is initially zero).

Given a correct implementation of the `Camera` class, the output from the application above should be *exactly* as shown below:

```
> java Q9
Camera on: true, photos taken: 0
Camera on: false, photos taken: 0
Camera on: true, photos taken: 7
Camera on: false, photos taken: 0
Camera on: false, photos taken: 7
Camera on: true, photos taken: 5
```

Notice that if a camera object is not turned on, then it can not take any photos.

SURNAME: ..... FORENAMES: .....

You need to define the `Camera` class used by the application `Q9`. Write your source code for this class below:

```
public class Camera {




}
```

(12 marks)

CONTINUED

SURNAME: ..... FORENAMES: .....

OVERFLOW PAGE

(If you have used this page, please indicate clearly under the relevant question that you have overflowed to this page)

SURNAME: ..... FORENAMES: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may detach this page from the answer booklet and use it for rough working)

SURNAME: ..... FORENAMES: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may detach this page from the answer booklet and use it for rough working)

SURNAME: ..... FORENAMES: .....

APPENDIX:

Useful methods and variables:

String

```
public int indexOf(char c)
public int indexOf(String string)
public char charAt(int index)
public String substring(int beginIndex, int endIndex)
public int length()
public boolean equals(String comparison)
```

StringTokenizer

```
public boolean hasMoreTokens()
public String nextToken()
```

Math

```
public static double random()
```