THE UNIVERSITY OF AUCKLAND

SUMMER SCHOOL, 2003

COMPUTER SCIENCE
Principles of Programming

TERMS TEST
(Time allowed: 75 MINUTES)

Surname:	
Forenames:	
Student ID number:	
Login name (UPI):	

INSTRUCTIONS:

- Attempt **ALL** questions - write your answers in the box provided
- Calculators are **NOT** permitted

Examiner to complete:

Question	Mark
1	(/15)
2	(/10)
3	(/3)
4	(/6)
5	(/6)
6	(/10)

Question	Mark
7	(/10)
8	(/10)
9	(/5)
10	(/10)
11	(/15)

TOTAL:	(/100)

CONTINUED

Question 1 (15 marks)

a) What is printed by the following?

```
System.out.println(5 / 2 * 2.5);
```

(1 mark)

b) What is printed by the following?

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

(2 marks)

c) What is printed by the following?

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

(2 marks)

d) What is printed by the following?

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

(2 marks)

e) What is printed by the following?

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

CONTINUED

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

(2 marks)

f) What is printed by the following?

```
System.out.println(Math.max(Math.min(9,10), Math.max(-12,7)));
```

(2 marks)

g) What is printed by the following?

```
String x = "Now";
String y = "Then";
System.out.println(x.length() + y.length());
```

(2 marks)

h) Complete the program below so that it prints out a single random integer to the screen. The random integer must be an **odd** number between 1 and 99 (inclusive). In other words, the output of the program must be either 1, 3, 5, 7, 9, 11, ..., 97 or 99:

```
public class OddInteger {

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

(2 marks)

```
     }
}
```

Question 2 (10 marks)

You need to locate and correct the errors in the application `Q2`. What the application is supposed to do is generate three random numbers between 0 and 9, print them to the screen, and also calculate and print the average of the three numbers to the screen.

For example, possible output when the *correct* application is run twice is given below:

```
C:/> java Q2
The average of: 4, 2 and 8 is 4.6666666666666

C:/> java Q2
The average of: 3, 9 and 3 is 5.0
```

The source code for the application `Q2` is given below, however it is not correct - it contains **four** syntax errors and **one** logic error.

You need to locate the syntax errors and the logic error in the source code below. For each error you must clearly **circle the error** and **also provide a correction**. You do not need to write out the whole line of source code again, as long as you indicate your correction clearly.

```
public Q2 {

     public static void main(String[] args) {

          int a;

          int b;

          it c;

          a = (int)(Math.random() * 10);

          b = (int)(Math.random() * 10);

          c = (int)(Math.random() * 10);

          double average = a + b + c / 3.0;

          System.out.print("The average of: ");

          System.out.print(a +  ", " b + " and " + c);

          System.out.println(" is " + average)

     }
}
```

(10 marks)

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

Question 3 (3 marks)

Using the values given for the variables below, evaluate the following two `boolean` expressions:

```
int x = 3;
int y = 5;
boolean isFalse = false;
```

(a) `!(x%2 == 0 || y%2 == 0)`

(1.5 marks)

(b) `(x<y && y>=5) || isFalse`

(1.5 marks)

Question 4 (6 marks)

What is the output when the following application is executed?

```
public class Q4 {

     public static void main(String[] args) {
          nestedIfs(5, 4);
     }

     private static void nestedIfs(int x,int y) {
          int number = x + y;
          if (x < y) {
               if (number > 6)
                    System.out.println("line 1");
               else if (number > 4)
                    System.out.println("line 2");
               else
                    System.out.println("line 3");
               System.out.println("line 4");
          }
          else {
               if (number > 6)
                    System.out.println("line 5");
               else if (number > 4)
                    System.out.println("line 6");
               else
                    System.out.println("line 7");
               System.out.println("line 8");
               number = number / 4;
          }
          if (number > 2)
               System.out.println("line 9");

          System.out.println("line 10");
     }
}
```

```
C:/> java Q4
```

(6 marks)

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

Question 5 (6 marks)

What is the output when the following application is executed?

```
public class Q5{

      public static void main(String[] args) {

            char c;
            String word = "GUNG HO";
            String newWord = "";
            int i = 1;
            int len = word.length();

            while (i < len){
                  c = word.charAt(i);
                  newWord = newWord + c;
                  System.out.println(newWord);
                  i = i + 2;
            }

      }

}
```

```
C:/> java Q5
```

(6 marks)

Question 6 (10 marks)

Deskcheck the following code using the technique of diagrams shown in lectures. Show both the diagram used to deskcheck the code (on facing page) and the output produced (below).

```
public class Q6 {

     public static void main(String[] args) {
          int num = 3;
          method1();
          System.out.println("main() num: " + num);
     }

     private static void method1() {
          int num = 5;
          num = method2(num);
          System.out.println("1. num: " + num);
     }

     private static int method2(int i) {
          int num = 5;
          String word = "Happy";
          word = method3(word, i);
          System.out.println("2. word: " + word);
          return (num + i);
     }

     private static String method3(String letters, int num1) {
          letters = letters.toUpperCase() + num1;
          System.out.println("3. letters: " + letters);
          return letters.substring(4);
     }
}
```

Show the output here:

```
C:/> java Q6
```

(4 marks)

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

Show the diagram here:

(6 marks)

Question 7 (10 marks)

Write the method `printNumbers()` which accepts two `int` parameters. The method prints all the numbers between the two `int`s passed in as parameters in ASCENDING order: i.e. if the first parameter is smaller than the second parameter, the numbers are printed from the first `int` parameter (inclusive) up to the second `int` parameter (inclusive), otherwise the numbers are printed from the second `int` parameter (inclusive) up to the first `int` parameter (inclusive).

For example, executing the `Q7` application (given below) with the completed `printNumbers()` method produces the following output:

```
C:/> java Q7

3 4 5 6 7 8

4 5 6 7 8 9

8

-2 -1 0 1 2 3
```

Carefully examine the source code for the application `Q7` below, and then complete the source code for the `printNumbers()` method on the following page:

```
public class Q7 {

     public static void main(String[] args){

          System.out.println();
          printNumbers(3, 8);

          System.out.println();
          printNumbers(9, 4);

          System.out.println();
          printNumbers(8, 8);

          System.out.println();
          printNumbers(3, -2);

     }
```

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

```
private static ______ printNumbers (______________________________){
```

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

```
  }
```

(10 marks)

```
}
```

Question 8 (10 marks)

Write the method `getEndDigits()` which accepts two parameters; the first parameter is a String made up entirely of digits e.g. “`3456723`” and the second parameter is an `int` representing the number of digits required. The method returns an `int`; it returns the required number of digits taken from the end of the String. For example, if the first parameter is “`3456723`” and the second parameter is `4`, then the integer made up of the last `4` digits, i.e. the `int 6723`, is returned by the method.

Note: if the number of digits required is more than the length of the String then the method returns the value `0`.

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

```
C:/> java Q8
3678
32
0
0
```

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

```
public class Q8 {

     public static void main(String[] args) {

          int number = getEndDigits("453678", 4);
          System.out.println(number);

          number = getEndDigits("8765432", 2);
          System.out.println(number);

          number = getEndDigits("32", 4);
          System.out.println(number);

          number = getEndDigits("", 4);
          System.out.println(number);
     }
```

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

```
private static ______ getEndDigits(________________________________){








 }
```

(10 marks)

```
}
```

Question 9 (5 marks)

What is the output when the following application is executed?

```
public class Q9 {

     public static void main(String[] args) {
          Point pt = new Point(6, 5);
          printPoint(pt);
          addToPoint(pt, 3);
          printPoint(pt);
     }

     private static void addToPoint(Point p, int addOn) {
          int x = (int) p.getX();
          int y = (int) p.getY();
          x = x + addOn;
          y = y + addOn;
          p.setLocation(x, y);
     }

     private static void printPoint(Point p) {
          int x = (int) p.getX();
          int y = (int) p.getY();
          System.out.println("point: " + x + "," + y);
     }
}
```

```
C:/> java Q9
```

(5 marks)

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

Question 10 (10 marks)

Complete the `numberTrue()` method below. This method is passed an array of `booleans` as a parameter, and must return the number of elements in the array which have the value `true`.

```
private static int numberTrue(boolean[] bools) {
```

(10 marks)

```
}
```

You can assume that the array of `booleans` passed to the method as the parameter, `bools`, contains at least one element when the method is called.

If you have written the method correctly, the code below:

```
boolean[] vals = {true, false, true, true, false, false, true};
System.out.println(numberTrue(vals));
```

should produce the following output:

```
4
```

Question 11 (15 marks)

Examine the source code for the following application carefully:

```
public class Q11 {

     public static void main(String[] args) {

          // Create the Car objects
          Car c1 = new Car("Mitsubishi", 0);
          Car c2 = new Car("Bentley", 50);
          Car c3 = new Car("Proton", 0);

          // Print the details of the Cars
          System.out.println(c1);
          System.out.println(c2);
          System.out.println(c3);
          System.out.println();

          // Change the speeds of the Cars
          c1.changeSpeed(90);
          c2.changeSpeed(-40);
          c3.changeSpeed(10);

          // Print the details of the Cars
          System.out.println(c1);
          System.out.println(c2);
          System.out.println(c3);
          System.out.println();

          // Compare the speeds of c2 and c3
          if (c2.sameSpeedAs(c3)) {
               System.out.print(c2.getName());
               System.out.print(" is moving the same speed as ");
               System.out.print(c3.getName());
          }
     }
}
```

The application `Q11` above uses a `Car` class to create three `Car` objects. When a `Car` object is constructed, the type of car (a String) and the current speed of the car (an `int`) are both specified. Given a correct implementation of the `Car` class, the output from the application above should be *exactly* as shown below:

```
     Mitsubishi not moving
     Bentley moving at 50 km/h
     Proton not moving

     Mitsubishi moving at 90 km/h
     Bentley moving at 10 km/h
     Proton moving at 10 km/h

     Bentley is moving the same speed as Proton
```

Notice that if the current speed of a car is `0`, the output “`not moving`” is displayed when the object is printed.

You need to define the `Car` class used by the application `Q11`. Write your source code for this class below:

```
public class Car {







}
```

(15 marks)

OVERFLOW PAGE

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

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

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)

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()
```

□□□□□□□□