THE UNIVERSITY OF AUCKLAND

SUMMER SEMESTER, 2008
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:	

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

CompSci 101 Test Results

Question	Marks	Out of
Question 1		26
Question 2		10
Question 3		10
Question 4		10
Question 5		6
Question 6		7
Question 7		7
Question 8		24
TOTAL		**100**

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

Question 1 (26 marks)

a) What output is produced by the following code?

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

(2 marks)

b) The following expression will evaluate to one of THREE different values. What are these THREE possible values?

```
int value = (int)(Math.random()*3) * 3 + 2;
```

(2 marks)

c) What would be printed by the following?

```
int x = Math.max(Math.max(4, 5), Math.min(6, 7));
System.out.println(x);
```

(2 marks)

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

d) What output is produced by the following code?

```
String a, b;

a = new String("cs101");
b = a;

a = b.substring(1,3);

System.out.println(a + b);
```

(2 marks)

e) What output is produced by the following code?

```
int number = 100;

while (number % 2 == 0) {
     number = number / 2;
     System.out.println(number);
}
```

(2 marks)

f) What would be printed by the following?

```
String a = "4";
int x = Integer.parseInt(a);
x = 10 + x;
System.out.println(x);
```

(2 marks)

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

g) What output is produced by the following code?

```
double x;

x = 1 / 2;

System.out.println(x);
```

(2 marks)

h) What output is produced by the following code?

```
int a = 1;
int b = 2;
int c = 3;

a = c;
c = b;
b = a;

System.out.println(a + " " + b + " " + c);
```

(2 marks)

i) What would be printed by the following?

```
System.out.println((1 + 2 * 3) / 6);
```

(2 marks)

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

j) Which of the following expressions does NOT evaluate to 7?

(a) `1 + 2 * 3`

(b) `(int)(6.6 + 0.5)`

(c) `7 % 10`

(d) `15 / 2`

(e) `(int)6.6 + (int)0.5`

Circle the letter of your answer:

a b c d e

(2 marks)

k) Consider the following code (notice the values assigned to variables `a`, `b` and `c` have been replaced with `???`):

```
int a = ???;
int b = ???;
int c = ???;

if (!(a <= b || b != c)) {
     System.out.println("success");
}
```

Give one set of possible values that could be assigned to `a`, `b` and `c` so that the word "`success`" will be printed:

```
int a =

int b =

int c =
```

(2 marks)

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

l) What output is produced by the following code?

```
int[] nums = {1, 3, 4, 7, 10};

int sum = 0;
for (int i = 1; i < nums.length; i++) {
     sum = sum + (nums[i] - nums[i-1]);
}

System.out.println(sum);
```

(2 marks)

m) What output is produced by the following code?

```
String a = new String("abc");
String b = new String("def");
String c = b;

System.out.println(a == c);
System.out.println(a.equals(c));
System.out.println(b == c);
System.out.println(b.equals(c));
```

(2 marks)

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

Question 2 (10 marks)

Consider the following code segment:

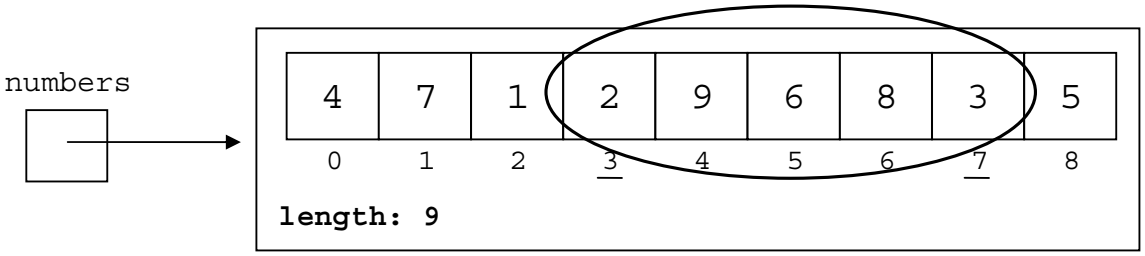
```
int[] numbers = {4,7,1,2,9,6,8,3,5};
int result = sumBetween(numbers, 3, 7);
System.out.println(result);
```

The output of this code is:

```
28
```

The `sumBetween()` method is passed an array, and two integers as parameters. The method calculates the sum of the elements in the array, starting at the index position specified by the first integer parameter, and ending at the index position specified by the second integer parameter.

In the example above, the starting index position is 3, and the ending index position is 7. The sum of the elements between (and including) these positions is 2 + 9 + 6 + 8 + 3 as illustrated in the diagram below. The result printed to the screen is therefore 28.



The following list of 12 statements is broken into 6 groups of 2 (i.e. 6 pairs). The correct definition of the `sumBetween()` method can be formed using exactly ONE statement from each pair. The pairs themselves are given in a random order, so you will need to select the correct statement from each pair as well as determine the correct order for the statements.

```
int sum = nums.length;
int sum = 0;

if (i >= start && i <= end)
if (i >= start || i <= end)

return sum;
return i;

private int sumBetween(int[] nums, int start, int end)
private void sumBetween(int[] nums, int start, int end)

for (int i = 0; i <= nums.length; i++)
for (int i = 0; i < nums.length; i++)

sum = sum + nums[i];
sum = nums[i];
```

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

In the space provided below, give the correct definition of the `sumBetween()` method. You must **only use the statements that were listed on the previous page**, but arranged in the correct order.

You will need to put **all of the opening and closing braces** into the code, and you must **indent** the code correctly.

(10 marks)

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

Question 3 (10 marks)

Consider the following program, which compares two Strings using a method called `charsMatchExactly()`:

```
public class MyProgram{

     public void start(){
          String s1 = "abc";
          String s2 = "cba";

          int matches = charsMatchExactly(s1, s2);

          System.out.println("Matches: " + matches);
     }


     private int charsMatchExactly(String a, String b) {

          int result = 0;

          if (a.charAt(0) == b.charAt(0)) {
               result++;
          }

          if (a.charAt(1) == b.charAt(1)) {
               result++;
          }

          if (a.charAt(2) == b.charAt(2)) {
               result++;
          }

          return result;
     }
}
```

The output from this program is:

```
Matches: 1
```

The `charsMatchExactly()` method calculates the number of characters which are exactly the same, and in exactly the same position in the two Strings which are passed to the method as parameters.

The limitation of the method is that it only works for Strings which have a length of 3 characters.

Rewrite the `charsMatchExactly()` method so that it correctly calculates the number of characters which are exactly the same, and in exactly the same position in the two Strings which can be of any length. You must use a loop to allow the method to work for Strings of arbitrary length. You can assume that the two Strings which are passed to the method as parameters are the same length.

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

```
private int charsMatchExactly(String a, String b) {









}
```

(10 marks)

If you complete the method correctly, the output of the following code:

```
String s1 = "abcdefgh";
String s2 = "cbadzzgz";

int matches = charsMatchExactly(s1, s2);

System.out.println("Matches: " + matches);
```

would be:

```
Matches: 3
```

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

Question 4 (10 marks)

The following program asks the user to enter a String at the command prompt, and then prints the String out to the screen, with one character removed. The character that is removed is selected at random from the characters in the String.

The following examples show the program running (user input is in bold):

```
C:\Test> java TestApplication
Enter word: hello
Output: ello

C:\Test> java TestApplication
Enter word: hello
Output: helo

C:\Test> java TestApplication
Enter word: hello
Output: hllo
```

The source code for this program is given below, but the definition of the `removeCharacter()` method is not provided. You must write the `removeCharacter()` method in the space provided below so that the program will produce the correct output as described above.

```
public class MyProgram {

     public void start() {
          System.out.print("Enter word: ");

          String word = Keyboard.readInput();
          String shorterWord = removeCharacter(word);
          System.out.println("Output: " + shorterWord);
     }
```

```
     private _____________ removeCharacter(_____________________) {









     }
```

(10 marks)

```
}
```

CONTINUED

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

Question 5 (6 marks)

What is the output of the following program?

```
public class MyProgram {

     public void start() {
          int x = 1;
          first(x + 1);
          System.out.println(x);
     }

     private void first(int x) {
          x = x + 1;
          x = x + second(x + 1);
          System.out.println(x);
     }

     private int second(int x) {
          System.out.println(x);
          return x + 1;
     }
}
```

(6 marks)

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

Question 6 (7 marks)

What is the output of the following program?

```
public class MyProgram {

     public void start() {

          int a = 2;
          int b = 3;

          a = first(a, b);
          b = first(b, a);

          System.out.println(a + " " + b);
     }

     private int first(int a, int b) {

          System.out.println(a + " " + b);

          a = a + 1;
          b = b - 1;
          return b * a;
     }
}
```

(7 marks)

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

Question 7 (7 marks)

What is the output of the following program?

```
public class MyProgram {

     public void start() {

          int[] y = {0, 1};
          int z = 1;

          changeArray(y);
          changeValue(z);

          System.out.println(y[0] + " " + y[1] + " " + z);
     }

     private void changeArray(int[] a) {
          a[0] = a[0] + 1;
          a[1] = a[1] + 1;

          System.out.println(a[0] + " " + a[1]);
     }

     private void changeValue(int a) {
          a = a + 1;

          System.out.println(a);
     }
}
```

(7 marks)

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

Question 8 (24 marks)

Below is the definition of the `Task` class:

```
public class Task {

     private String description;
     private String personWhoDidTask;
     private boolean hasBeenDone;
     private int numberOfMinutes;

     public Task(String description, int minutes) {
          this.description = description;
          numberOfMinutes = minutes;
          hasBeenDone = false;
          personWhoDidTask = "";
     }

     public void setPersonWhoDidTask(String name) {
          hasBeenDone = true;
          personWhoDidTask = name;
     }

     public String getPersonWhoDidTask() {
          if (hasBeenDone) {
               return personWhoDidTask;
          }
          return "task has not been done";
     }
```

```
     public _________________ setNumberOfMinutes( _________________ ) {




     }
```

(3 marks)

```
     public _________________ getNumberOfMinutes() {




     }
```

(3 marks)

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

```
    public String toString() {

        String info = description + " ";
        info = info + numberOfMinutes;

        if (numberOfMinutes > 1) {
            info = info + " minutes";
        } else {
            info = info + " minute";
        }
        if (hasBeenDone) {
            info = info + ", done by " + personWhoDidTask;
        }
        return info;
    }

    public boolean takesLongerThan(Task other) {
        if (numberOfMinutes > other.numberOfMinutes) {
            return true;
        }
        return false;
    }
}
```

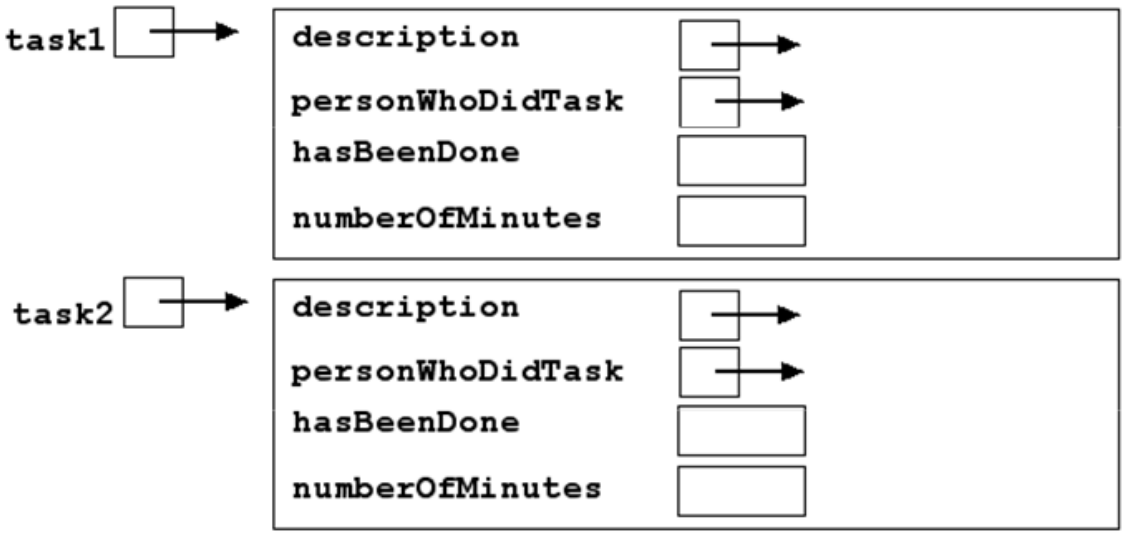
a) In the `Task` class, in the spaces provided, complete the following TWO method definitions:

- the accessor method, `getNumberOfMinutes()`
- the mutator method, `setNumberOfMinutes()`

b) Two objects of type Task are created as follows:

```
Task task1 = new Task("Clean bathroom", 10);
Task task2 = new Task("Take out rubbish", 1);
```

Complete the diagram below illustrating the values that are stored in the instance variables for each of these objects. You should write **very clearly** on the diagram, as you are required to change the diagram when you complete part (c) of this question.



(9 marks)

CONTINUED

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

c) Given the two `Task` objects created and initialised as in the diagram above, give the output when the following statements are executed. **Note: you MUST also mark any changes to any instance variables clearly on the diagram above.**

```
System.out.println("1 " + task1.toString());
System.out.println("2 " + task2.toString());

task1.setPersonWhoDidTask("Tom");
System.out.println("3 " + task1.toString());

String name = task2.getPersonWhoDidTask();
System.out.println("4 " + name);

task2.setPersonWhoDidTask("Adriana");
System.out.println("5 " + task2.toString());

if (task2.takesLongerThan(task1)) {
     System.out.println("6 ");
} else {
     System.out.println("7 ");
}
```

The output is:

(9 marks)

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)