THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2002
Campus: City and Tamaki

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	SOLUTIONS
Forenames	
Student ID	
Laboratory Day and Time	

Question	Marks	Out of
Question 1 (Primitive Types)		**15**
Question 2 (Strings)		**10**
Question 3 (If statements)		**10**
Question 4 (Assignment 02)		**10**
Question 5 (Assignment 02)		**10**
Question 6 (Loops)		**8**
Question 7 (Arrays)		**12**
Question 8 (Classes)		**15**
Question 9 (Frames)		**10**
Total		**100**

This page is left blank

1.   What is the output for each of the following statements?

(a)   `System.out.println( 1 + 2 + "3" + "4" + 5 + 6);`

33456

(3 marks)

(b)   `System.out.println( 2/3 + 3.0/2.0 + " " + 2/3 + 3.0/2.0);`

1.5 01.5

(3 marks)

(c)   `System.out.println( 5%10 + " " + 10%3 );`

5 1

(3 marks)

(d)   `System.out.println( 1* 2+3 %4 +5/ 6+1.0 );`

6.0

(3 marks)

(e)   `System.out.println( "\"+"+"\"+\n^\\Test\\" );`

"+"+
^\Test\

(3 marks)

2.   Write a method called `removeLetter` which accepts a `String` `s` and an integer `i` as parameters. Your method should remove a single character from the `String` passed as a parameter, and it should return the `String` with the character removed. The character which is to be removed is located at the index specified by `i`. Example:

Executing the statements:

```
System.out.println( removeLetter( "Hello", 0 ) );
System.out.println( removeLetter( "Hello", 1 ) );
System.out.println( removeLetter( "Hello", 2 ) );
```

would produce the output:

```
ello
Hllo
Helo
```

You can assume that the integer passed to the method as the index will be a valid index value (e.g. it will be in the range 0 … length-1). Write your method below.

```
    private static String removeLetter( String s, int i ){
            String result = s.substring(0,i) + s.substring(i+1);
            return result;
    }
```

Note: No marks are allocated to private or static.

(10 marks)

3. What is the output of each of the following code fragments?

(a)   `System.out.println( 23==34 || 3==3);`

true

(2 marks)

(b)   `System.out.println( 1!=2 && !(3==3) || 2==2 );`

true

(2 marks)

(c)
```
int a = 4;
int b = 4;
int c = 2;

if( a==b )
   if ( b<c ){
       if ( c>a )
           System.out.println( "First" );
   }
   else
       System.out.println( "Second" );
else
   System.out.println( "Third" );
System.out.println( "Fourth" );
```

Second
Fourth

(2 marks)

(d)
```
int a = 0;
int b = 3;
if( !(a<b) || (a!=b && b==3) )
System.out.println( a );
if( !(a<b || a!=b && b==3) )
System.out.println( b );
System.out.println("done");
```

0
done

(4 marks)

4.   Using a table containing two lists of letters, we can convert one letter to another.

Alphabetical List	Scrambled List
A	X
B	J
C	Q
D	F
E	B
…	…

In order to convert any given letter, we would look up the letter in the Alphabetical List, then simply find the corresponding letter in the same position in the Scrambled List.

Write a method which will convert a single character in the manner described above. Your method should accept two `Strings` and a `char` as parameters. The first `String` contains the Alphabetical List. The second `String` contains the Scrambled List. Your method should return the single character which is obtained by looking up the original character in the Alphabetical List and finding the corresponding character in the Scrambled List.

Example:
```
    convertLetter(  "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
                    "ZXVTRPNLJHFDBAYWUSQOMKIGEC", 'H' );
```
would return the character:
    L

You can assume that the character you are converting will always be an upper case letter (so it will always exist in the Alphabetical List). The following method header has been provided for you:

```
private static char convertLetter(    String alphabetical,
                                      String scrambled,
                                      char c  ){



    int index = alphabetical.indexOf(c);

    char result = scrambled.charAt(index);

    return result;




}
```

(10 marks)

5.   Write a method which converts an entire word to a scrambled version using the technique described in question 4 above. You can assume that the `convertLetter()` method you wrote for question 4 works correctly, and you should call that method to convert each character in the `String`. Note that you will have to use a loop to convert each character in the original word, one letter at a time. Your method should accept a single word as a parameter and it should return the converted word.

Your method will contain at least two variables (these are provided for you). The first variable is a `String` containing the list of letters in Alphabetic order, and the second variable is a `String` containing the list of letters in Scrambled order. You can use these variables when you make any method calls to the `convertLetter()` method.

You can assume that the word passed as a parameter will contain only upper case letters from the alphabet (in other words, you do not have to worry about the space character, or other letters which do not appear in the list of letters).

Example:
```
    convertWord( "HELLO" );
```

would return the word:
```
    LRDDY
```

The following skeleton code has been provided for you:

```
private static String convertWord( String word ){
     String alphabetic   = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
     String scrambled    = "ZXVTRPNLJHFDBAYWUSQOMKIGEC";


     String result = "";

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

          char letter = word.charAt(i);

          char converted = convertLetter(alphabetical, scrambled, letter);

          result = result + converted;

     }


     return result;




}
private static char convertLetter( String a, String s, char c){
     //assume the code for this method works correctly
     ...
}
```

(10 marks)

6.   What is the output when the Q6 application is executed?

```
public class Q6{
    public static void main(String[] args){
        for(int i=50; i<100; i=i+10){
            if (i%20 == 0)
                System.out.println("i=" + i + ": " + (i+2));
            else
                System.out.println("i=" + i + ": " + (i-2));
        }
    }
}
```

```
>java Q6

i=50: 48
i=60: 62
i=70: 68
i=80: 82
i=90: 88



```

(8 marks)

7.   Complete a method `accumArray()` which is passed one parameter, an array of integers. This method returns a new `int` array which contains the cumulative sum of all the previous elements (including the current element) of the parameter array.

For example, if the parameter array contains the integer values:
```
10 20 30 10
```
then the `accumArray()` method will return a new array which contains the integer values:
```
10 30 60 70.
```

As a further example if the parameter array contains the integer values:
```
5 20 15 10 5 5
```
then the `accumArray()` method will return a new array which contains the integer values:
```
5 25 40 50 55 60
```

```java
private static int[] accumArray(int[] intArray){

    int[] returnA = new int[intArray.length];
    int upTo = 0;

    for(int i=0; i<intArray.length; i++){
        upTo = upTo + intArray[i];
        returnA[i] = upTo;
    }

    return returnA;

}
```

(12 marks)

8.   Study the definition of the `Run` class shown below.

```
public class Run{
     private static int nextID = 1000;
     private int runID;                     //ID number of the run
     private int numKms;                    //number of kilometres run
     private int numMins;                        //number of minutes

     public Run(int kms, int mins){
          runID = nextID;
          nextID++;
          numKms = kms;
          numMins = mins;
     }

     public int getRunID(){
          return runID;
     }

     public void setNumKms(int kms){
          numKms = kms;
     }

     public void setNumMins(int mins){
          numMins = mins;
     }

     public String toString(){
          return "Run id: "+runID+ ", Kilometres: "+numKms+
                                       ", Minutes: "+numMins;
     }

     public boolean isFasterThan(Run otherR){
          return (double)numMins/numKms <
                         (double)otherR.numMins/otherR.numKms;
     }
}
```
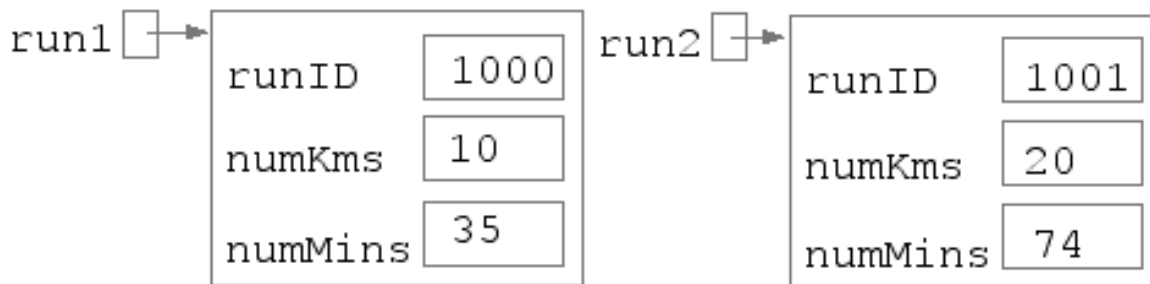
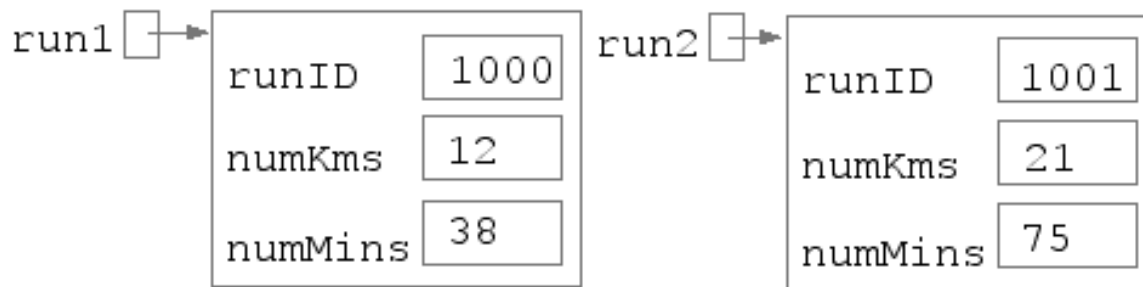Write the code for questions a), b), c) and d) in the spaces provided.

a)   Write code to create 2 `Run` instances which look like the following in memory:


```
   Run run1, run2;

   run1 = new Run(10,35);    //2 marks
   run2 = new Run(20,74);
```

b) Write code to change [do NOT create new instances] the 2 Run instances so they now look like the following in memory:


```
    run1.setNumKms(12);           //4 marks
    run1.setNumMins(38);
    run2.setNumKms(21);
    run2.setNumMins(75);
```

c) Write code using the 2 Run instances which will give the following output:

```
    Run1: Run id: 1000, Kilometres: 12, Minutes: 38
    Run2: Run id: 1001, Kilometres: 21, Minutes: 75
```

```
                                       //4 marks
System.out.println("Run1: "+run1.toString());
System.out.println("Run2: "+run2.toString());
```

d) Write code which tests which of the 2 Run instances is faster. If run1 is faster the output will be:

```
    Run 1000 is faster
```

If run2 is faster the output will be:

```
    Run 1001 is faster
```

Note: The runID value shown in the output to part d) is obtained from the Run instance. No message is printed if the two runs are of equal speed.

```
                                       //5 marks
if (run1.isFasterThan(run2))
    System.out.println("Run "+run1.getRunID()+ " is faster");
else if(run2.isFasterThan(run1))
    System.out.println("Run "+run2.getRunID()+ " is faster");
```

(15 marks)

9.   The `Q9` application shown below creates a `Q9Frame` instance.

```
import java.awt.*;

public class Q9{
    public static void main(String[] args){
        Q9Frame theFrame = new Q9Frame("Question 9", 10, 20, 190, 200);
    }
}
```

Complete the `Q9Frame` class definition on the next page so that when the `Q9` application is executed the following `Frame` is displayed. I have added some notes below the `Frame` to indicate the position of the line and shapes.
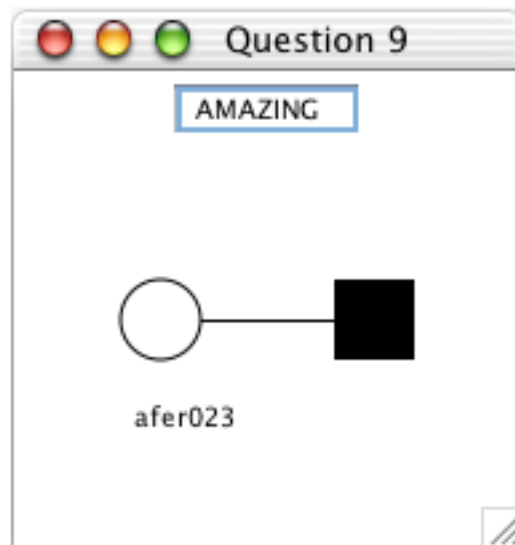
The circle has a top left position at (40,100).
The square has a top left position at (120,100).
The circle has a radius of 15 pixels (diameter 30).
The square is of size 30 pixels.
The line between the circle and the square is 50 pixels long.

The `String` "afer023" is at position (45,155); The `String` displayed inside the `Frame` should be your own upi (login name).

The `TextField` contains the string "AMAZING".

```java
import java.awt.*;
import java.awt.event.*;

public class Q9Frame    extends Frame    {
    private TextField theTextF;

    public Q9Frame(String title, int top, int left,
                            int width, int height){
        setTitle(title);
        setBounds(top,left, width, height);
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e){
                dispose();
                System.exit(0);
            }});
        setLayout(new FlowLayout());
```

```java
                    //3 marks
    TextField theTextF = new TextField(" AMAZING ");
    add(theTextF);
```

```java
        show();
    }

    public void paint(Graphics g){
```

```java
           //6 marks
 g.drawOval(40,100,30,30);
 g.fillRect(120,100,30,30);
 g.drawLine(70,115,120,115);
  //Their upi in the drawstring() method
 g.drawString("afer023",45,155);
```

```java
    }
}
```

(10 marks)

Overflow Sheet

Write the question number next to your answer.
You must **ALSO** indicate in the allotted space that you have used the overflow sheet.

_____

Rough Working

You may remove this page from the exam script if you wish. This page will not be marked.

Rough Working

You may remove this page from the exam script if you wish. This page will not be marked.