THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2001
Campus: City/Tamaki

COMPUTER SCIENCE

TEST SOLUTIONS

Principles of Programming

(Time allowed: ONE hour and 15 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	
Login (UPI)	
Lab Day and Time	

Section	Marks	Out of
SECTION A		**20**
SECTION B		**30**
SECTION C		**10**
Total		**60**

SECTION A

1. Examine each of the following Java statements carefully. For each of the statements, tick the appropriate box to indicate either that the statement is correct, or if it contains any errors, then indicate the category of error (Syntax, Runtime or Logic). If the statement contains an error, then correct the error by writing a correct version of the statement below the original. (Note that there may be more than one way to correct an error - any sensible correction will fine)

(a) `int i = 34;`

Correct ☑ Syntax Error ☐ Runtime Error ☐ Logic Error ☐

(b) `int i = 3.0 + 5;`

Correct ☐ Syntax Error ☑ Runtime Error ☐ Logic Error ☐

```
int i = 3 + 5; or double i = 3.0 + 5;
```

(c) `double customer2345 = 87.754;`

Correct ☑ Syntax Error ☐ Runtime Error ☐ Logic Error ☐

(d) `double 2ndAverage = 9.08;`

Correct ☐ Syntax Error ☑ Runtime Error ☐ Logic Error ☐

```
double secondAverage = 9.08;
```

(e) `double d = 45`

Correct ☐ Syntax Error ☑ Runtime Error ☐ Logic Error ☐

```
double d = 45;
```

(f) `int second = Integer.parseInt("87b");`

Correct ☐ Syntax Error ☐ Runtime Error ☑ Logic Error ☐

```
int second = Integer.parseInt("87");
```

(g) `String s = 98;`

Correct ☐ Syntax Error ☑ Runtime Error ☐ Logic Error ☐

```
String s = "98";
```

(h) `String u == "hello" + "world";`

Correct ☐ Syntax Error ☑ Runtime Error ☐ Logic Error ☐

```
String u = "hello" + "world";
```

(i) `System.out.println(3+4+"is the sum of 3 and 4");`

Correct ☑ Syntax Error ☐ Runtime Error ☐ Logic Error ☐

(j) `System.out.println("(3 + 4) * 7 = " + (3 + 4) * 7);`

Correct ☑ Syntax Error ☐ Runtime Error ☐ Logic Error ☐

(10 marks)

Marking Notes (Andrew):

- *A Syntax error is generated when the code is compiled*
- *A Runtime error means that the code compiles correctly, but error occurs when code is executed*
- *A Logic error means that the code compiles and executes, but does not produce the desired result.*

*Since this questions is about possible errors, **any** error in written code for this question will result in marks lost.*

2. What is the output of the following application?

```
public class Q2{
     private static void a(){
            System.out.println("a");
     }
     private static void b(){
            System.out.println("b");
            a();
     }
     private static void c(){
            System.out.println("c");
            b();
            a();
     }
     public static void main(String[] args){
            System.out.println("Started");
            a();
            b();
            c();
            System.out.println("Finished");
     }
}
```

```
Started
a
b
a
c
b
a
a
Finished
```

(5 marks)

Marking Notes (Andrew):

- *The majority of the marks are allocated to method c()*

3. Write a method called upperBound() that takes 3 doubles as parameters and returns the smallest integer number which is **greater than** all 3 input values. Your method should be declared as `static` and `private`.

Example: If the method was passed -23.4, 0.0003, and 27.3 as parameters, then it would return the int value 28.

```
public class Q3{

	//Write your method here:


	private static int upperBound(double a, double b, double c){
		double first = Math.max(a,b);
		double max = Math.max(first, c);
		return (int)(max +1);
	}




	public static void main(String[] args){
		int result = upperBound(-23.4, 0.0003, 27.3);
		System.out.println(result);
	}
}
```

(5 marks)

Marking Notes (Say Gin):

- *Partial marks may be awarded for code which is partially correct (e.g. contains syntax errors etc.)*

SECTION B

4. What is the output produced when the following application is executed?

```
public class Q4{
	public static void main(String[] args){
		int num1 = 15;
		int num2 = 34;
		int num3;

		if (num1>4 && num2<22){
			num3 = num2-num1;
			if (num3>30)
				System.out.println("output 1");
			else if (num3>20)
				System.out.println("output 2");
			else
				System.out.println("output 3");
			System.out.println("output 4");
		}
		else {
			num3 = num2+num1;
			if (num3<30)
				System.out.println("output 5");
			else if (num3<50 || num3%2==1)
				System.out.println("output 6");
			else
				System.out.println("output 7");
			System.out.println("output 8");
		}
	}
}
```

```
output 6
output 8
```

(4 marks)

5. The application, Q5, uses a class called `DigitSum` (the definition of the `DigitSum` class is shown on the next page). What is the output when the application Q5 is executed?

```
public class Q5{

	public static void main(String[] args){
		DigitSum dig1;

		dig1 = new DigitSum(345);
		System.out.println(dig1.toString());

		dig1 = new DigitSum(2001);
		System.out.println(dig1.toString());
	}

}
```

```
Number: 345
Sum of Digits: 12
Number: 1000
Sum of Digits: 1
```

(5 marks)

```
/*
      This code defines the class DigitSum
*/

public class DigitSum{

      private int number,sumDigits;

      public DigitSum(int num){
            if (num < 0)
                  number = 0;
            else if (num > 1000)
                  number = 1000;
            else
                  number = num;
            sumDigits = workOutSumDigits();
      }

      private int workOutSumDigits(){
            //works out the sum of the individual digits in the number
            int sum = 0;
            String stringN = "" + number;

            while(stringN.length() > 0){
                  sum = sum + Integer.parseInt(stringN.substring(0,1));
                  stringN = stringN.substring(1,stringN.length());
            }

            return sum;
      }

      public String toString(){
            String tempStr= "Number: " + number;
            tempStr += "\nSum of Digits: " + sumDigits;
            return tempStr;
      }
}
```

6. What is the output of the following application?

```
public class Q6{
	public static void main(String[] args){
		String upS = "";
		int count = 0;
		while (count < 5){
			if (count%2 == 0)
				upS = upS + ":"+count;
			else
				upS = upS + "-"+count;

			System.out.println(upS);
			count++;
		}
		System.out.println();
	}
}
```

```
:0
:0-1
:0-1:2
:0-1:2-3
:0-1:2-3:4
```

(4 marks)

Marking Notes(Angela):

- *Note that this question is worth 4 marks… some scripts have a typo that it is worth 3.*
- *Ignore final blank line*

7. The following application uses a class called Appointment:

```
public class Q7{
	public static void main(String[] args){
		Appointment app1, app2, app3;
		app1 = new Appointment("Bill Bro","Hick",30);
		app2 = new Appointment("Kim Strong","Sook",45);
		app3 = new Appointment("Jane Spleen","Hick",15);
		app2.setTime(25);
		app1.setIsPaid(true);
		app1.setTime(60);
		app3.setIsPaid(true);
		System.out.println(app1.toString());
		System.out.println("************************************");
		System.out.println(app2.toString());
		System.out.println("************************************");
		System.out.println(app3.toString());
		System.out.println("************************************");
	}
}
```

The output when I execute the above application is shown below:

```
> java Q7
PATIENT: Bill Bro, Dr. Hick
APPOINTMENT LENGTH: 60 minutes.  HAS BEEN PAID
**********************************************
PATIENT: Kim Strong, Dr. Sook
APPOINTMENT LENGTH: 25 minutes. NOT PAID
**********************************************
PATIENT: Jane Spleen, Dr. Hick
APPOINTMENT LENGTH: 15 minutes.  HAS BEEN PAID
```

Write the `Appointment` class definition:

```
public class Appointment{
  private int appLength;
  private String patient;
  private String doctor;
  boolean isPaid;

  public Appointment(String pName,String dName,int lenTime){
   patient = pName;
   doctor = dName;
   appLength = lenTime;
   isPaid = false;
  }

  public void setTime(int newLength){
   appLength = newLength;
  }

  public void setIsPaid(boolean newIsPaid){
   isPaid = newIsPaid;
  }

  public String toString(){
   String tempStr= "";
   tempStr += "PATIENT: "+patient+", Dr. "+doctor;
   tempStr += "\nAPPOINTMENT LENGTH: "+ appLength+" minutes. ";
   if (isPaid)
     tempStr += " HAS BEEN PAID";
   else
      tempStr += "NOT PAID";
   return tempStr;
  }

}
```

(10 marks)

8. Study the following class definition:

```
public class Word{

	private static int number = 0;
	private String theWord;
	private int value;

	public Word(String word, int val){
		theWord = word;
		value = val;
		number = number + val;
	}

	public void setValue(int val){
		number = number - value;
		value = val;
		number = number + value;
	}

	public void display(){
		System.out.println(value + " - " + theWord + " - " + number);
	}

	public static void setNumber(int num){
		number = num;
	}
}
```

What is the output when the following application is executed?

```
import java.util.*;

public class Q8{

	public static void main(String[] args){
		Word word1, word2, word3;

		Word.setNumber(100);

		word1 = new Word("Press", 2);
		word1.display();

		word2 = new Word("Intention", 4);
		word3 = new Word("Enigmatic",7);

		word2.setValue(5);
		word2.display();

		word3.setValue(6);
		word3.display();
	}
}
```

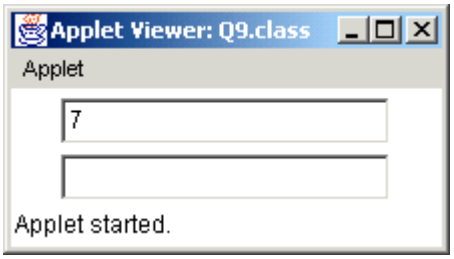
```
2 - Press - 102
5 - Intention - 114
6 - Enigmatic - 113
```

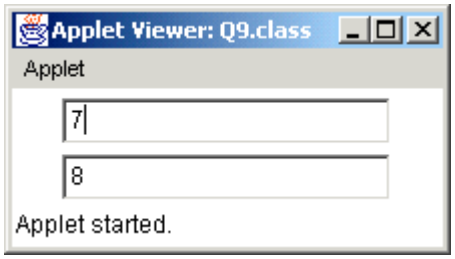
(7 marks)

SECTION C

9. Complete the `actionPerformed()` method of the following applet. When a value is entered in the top `TextField`, the bottom `TextField` should display the value which is one larger than the value entered.

Screenshots included here show an example of the applet with a value entered into the first `TextField`, and the result produced in the second `TextField` when the enter key is pressed.






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

public class Q9 extends Applet implements ActionListener {

      private TextField tInput;
      private TextField tOutput;

      public void init() {
            tInput = new TextField(20);
            tOutput = new TextField(20);
            tInput.addActionListener(this);
            add(tInput);
            add(tOutput);
      }

      public void actionPerformed(ActionEvent e) {

            int value = Integer.parseInt(tInput.getText());
            tOutput.setText("" + (value + 1));

      }
}
```

(5 marks)

Q10 Complete the sum() method below. It is passed an array of `ints` as a parameter and returns the sum of all the values in the array.

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

public class Q10 extends Applet {

      public int sum(int[] values) {

            int result = 0;
            for (int i = 0; i < values.length; i++)
                  result += values[i];
            return result;

      }

      public void init() {
            int[] numbers = {4, 6, 2, 8, 3, 1};
            int result = sum(numbers);
            System.out.println("The sum is: " + result);
      }

}
```

If the sum method is written correctly, then the applet above will produce 24 as output (given the numbers chosen in the array).

(5 marks)

Overflow Sheet 1

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

Overflow Sheet 2

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