THE UNIVERSITY OF AUCKLAND

SUMMER SEMESTER, 2007
Campus: City

COMPUTER SCIENCE

Principles of Programming

(Time allowed: TWO hours)

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:	

Q1 (/20)	**Q4** (/6)	**Q7** (/13)	**Q10** (/6)
Q2 (/8)	**Q5** (/8)	**Q8** (/10)	**TOTAL**
Q3 (/5)	**Q6** (/11)	**Q9** (/13)	
			(/100)

Question/Answer Sheet ID: .....

Question 1 (20 marks)

a) Assume that the current directory contains a Java source file named `MyProgram.java`. The following DOS command is used to compile the source file:

```
javac MyProgram.java
```

If the file compiles successfully, what is the name of the new file that will be created?

MyProgram.class

(2 marks)

b) One of the lines of code in the following program is surrounded by a comment. What is the output of the program?

```
public class MyProgram{

     public void start(){

          int x = 100;

          if (x > 50) {
               System.out.println("A");
          /* } else { */
               System.out.println("B");
          }
     }
}
```

A
B

(2 marks)

Question/Answer Sheet ID: .....

c) The `positive()` method is *called* in the following way:

```
int[] x = {1,2,3,4,5};
int[] nums = positive(x, true);
```

Complete the method header for the `positive()` method (i.e. complete the first line of the method definition).

```
private int[] positive(int[] nums, boolean b) {
     for (int i = 0; i < nums.length; i++) {
          if ((nums[i] < 0) && (b)) {
               nums[i] = -nums[i];
          }
     }
     return nums;
}
```

(2 marks)

d) What is the output of the following code?

```
String a = new String("hello");
String b = new String("hello");
String c;

c = a;

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

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

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

```
true
false
true
true
true
false
```

(2 marks)

Question/Answer Sheet ID: .....

e) What is the output of the following code?

```
int[] nums = {1, -2, 3, -4, 5};

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

for (int i = 0; i < nums.length; i++) {
     System.out.print(nums[i] +  " ");
}
```

1 2 3 4 5

(2 marks)

f) What is the output of the following code?

```
String a = "123";

System.out.println(a + 1);
System.out.println(Integer.parseInt(a) + 1);
```

1231
124

(2 marks)

g) What is the output produced by the following code?

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

20

(2 marks)

Question/Answer Sheet ID: .....

h) Write a Java statement which declares and constructs an array of 30 String objects. The array should be called `names`.

```
String[] names = new String[30];
```

(2 marks)

i) What is printed by the following code segment?

```
String[] s = { "how", "when", "why" };
System.out.println(s[1].length());
```

```
4
```

(2 marks)

j) What is printed by the following code segment?

```
String[] s = { "zero", "one", "two" };

s[1] = s[2];
s[0] = s[1];

for(int i = 0; i < s.length; i++) {
     System.out.print(s[i] + " ");
}
```

```
two two two
```

(2 marks)

Question/Answer Sheet ID: .....

Question 2 (8 marks):

Complete the `printTime()` method below which is passed a total number of seconds (which may be larger than 60) as a parameter and prints the equivalent number of minutes and seconds (where the number of seconds must be less than 60).

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

```
printTime(15);
printTime(60);
printTime(135);
```

should be:

```
0 minutes and 15 seconds
1 minutes and 0 seconds
2 minutes and 15 seconds
```

Complete the `printTime()` method in the space provided below:

```
private void printTime(int totalSeconds) {
     int numberOfSeconds, numberOfMinutes;

     numberOfMinutes = totalSeconds / 60;
     numberOfSeconds = totalSeconds % 60;

     System.out.print(numberOfMinutes + " minutes");
     System.out.print(" and ");
     System.out.println(numberOfSeconds + " seconds");
}
```

(8 marks)

Question/Answer Sheet ID: .....

Question 3 (5 marks)

Consider the following method called `process()` which is passed an array of `int`s as a parameter:

```
private int process(int[] values) {
     int result = values[0];
     for (int i = 1; i < values.length; i++) {
          if (values[i] < result) {
               result = values[i];
          }
     }
     return result;
}
```

Describe in words what *value* the method `process()` returns for any given array.

The smallest element in the array

(3 marks)

Now, assume that the `process()` method is called from the following `start()` method, which passes it an array of 1,000,000 integers with random values between 0 and 999 (inclusive):

```
public void start() {

     int[] numbers = new int[1000000];
     for (int i = 0; i < numbers.length; i++) {
          numbers[i] = (int)(Math.random()*1000);
     }

     int result = process(numbers);

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

What is the *most likely* value to be printed by the `start()` method?

0

(2 marks)

Question/Answer Sheet ID: .....

Question/Answer Sheet ID: .....

Question 4 (6 marks)

Complete the output when the following code is executed.

```
Point pt1, pt2, pt3;

pt1 = new Point(20, 30);
pt2 = new Point(20, 30);
pt3 = pt2;
System.out.println("1.  " + (pt1 == pt2));

pt1.move(5, 20);
System.out.println("2.  " + pt1.x + "," + pt1.y);

pt2.translate(5, 20);
System.out.println("3.  " + pt2.x + "," + pt2.y);

pt3.x = pt1.y;
pt3.y = pt1.x;
System.out.println("4.  " + pt3.x + "," + pt3.y);
System.out.println("5.  " + pt1.x + "," + pt1.y);

System.out.println("6.  " + (pt1.equals(pt2)));
```

```
1.  false

2.  5,20

3.  25,50

4.  20,5

5.  5,20

6.  false
```

(6 marks)

Question/Answer Sheet ID: .....

Question 5 (8 marks)

Draw the output produced by the following code in the grid given at the bottom of the page.

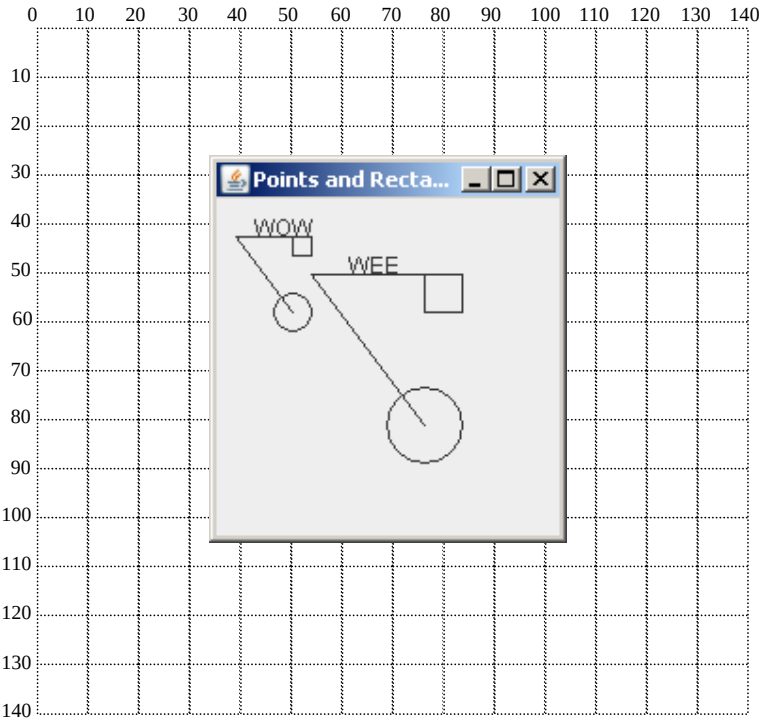
The grid lines are not part of the output but are there to help you place the drawing in the correct position. The size of each square in the grid is 10 pixels by 10 pixels.

```
import javax.swing.*;
import java.awt.*;

public class A1JPanel extends JPanel {

     public void paintComponent(Graphics g) {
          super.paintComponent(g);
          draw(g, 10, 20, 10, "WOW");
          draw(g, 50, 40, 20, "WEE");
     }

     private void draw(Graphics g, int x, int y, int size,
                                                  String message) {
          g.drawLine(x, y, x + size*4, y);
          g.drawLine(x, y, x + size*3, y+size*4);
          g.drawRect(x + size*3, y, size, size);
          g.drawOval(x + size*2, y+size*3, size*2, size*2);
          g.drawString(message, x + size, y);
     }
}
```



(8 marks)

Question/Answer Sheet ID: .....

Question 6 (11 marks)

The `JPanel` defined on the next page contains two `JLabel` components, three `JTextField` components and one `JButton` component:

- `centsPerKL` – a JLabel displaying "Cents per kilometre"
- `centsPerKT` – a JTextField which contains a whole number representing the cost per kilometre
- `numberKL` – a JLabel displaying "Kilometres"
- `numberKT` – a JTextField which contains a whole number representing the number of kilometres
- `priceB` – a JButton displaying the String, "PRICE".
- `priceT` – a JTextField which displays, in dollars and cents, the cost of travelling the stated number of kilometres

Below is a screenshot of the `JPanel` when it is first displayed. The `centsPerKT` JTextField displays "`12`", the `numberKT` JTextField displays "`100`" and the `priceT` JTextField displays "`$12.0`".

In the `centsPerKT` JTextField, the user enters a whole number, representing the cost in cents of travelling one kilometre. In the `numberKT` JTextField, the user enters a whole number, representing the number of kilometres. When the user presses the `PRICE` JButton, the cost of travelling the stated number of kilometres at the stated cost in cents per kilometre is displayed, in dollars and cents, in the `priceT` JTextField.

You are required to complete the `JPanel` definition on the next page so that the `JPanel` behaves as described above. You must use the variables given in the code.

The screenshots below show the `JPanel` after the user has pressed the `PRICE` JButton.

Question/Answer Sheet ID: .....

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

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

```
import javax.swing.*;

public class AJPanel extends JPanel    implements ActionListener  {

     private JTextField numberKT, centsPerKT, priceT;
     private JButton priceB;

     public AJPanel() {
          JLabel numberKL = new JLabel("Kilometres");
          JLabel centsPerKL = new JLabel("Cents per kilometre");
          numberKT = new JTextField(4);
          centsPerKT = new JTextField(4);
          priceT = new JTextField(12);
          priceB = new JButton("PRICE");
```

```
          priceB.addActionListener(this);
```

```
          centsPerKT.setText("12");
          numberKT.setText("100");
          priceT.setText("$12.0");

          add(centsPerKL);
          add(centsPerKT);
          add(numberKL);
          add(numberKT);
          add(priceB);
          add(priceT);
     }
```

```
     public void   actionPerformed  (  ActionEvent   e) {


          int centsPerK = Integer.parseInt(centsPerKT.getText());
          int ks = Integer.parseInt(numberKT.getText());

          double cost = centsPerK * ks / 100.0;

          priceT.setText("$" + cost);









     }
}
```

(11 marks)

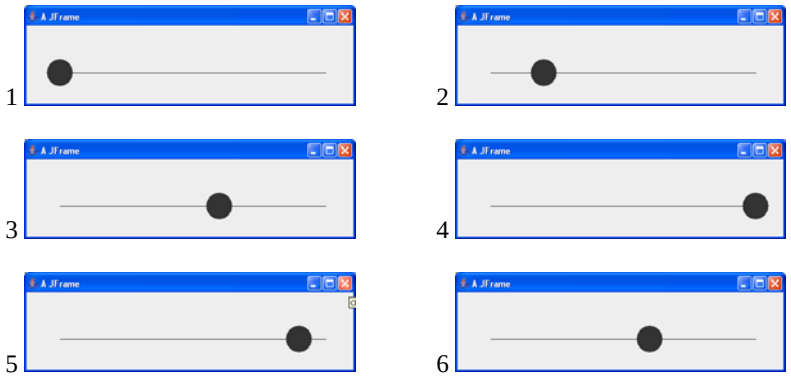
CONTINUED

Question/Answer Sheet ID: .....

Question 7 (13 marks)

The following `JPanel` uses a `Timer` object. The `JPanel` displays a circle which moves continuously along a horizontal line from left to right and back again. Once the `JPanel` is in focus, the user controls when the ball starts and stops moving by pressing the UP and DOWN arrow keys. When the user presses the UP arrow key the `Timer` should start and the ball begins to move from its current position in whichever direction it is currently moving. When the user presses the DOWN arrow key the `Timer` should stop and the ball stops in its current position.

Below are some screenshots of the `JPanel` in action. Screenshot 1 shows the `JPanel` when it is first displayed. The ball is not moving. Screenshots 2, 3 and 4 show the `JPanel` after the user has pressed the UP arrow key and the ball is moving to the right. Screenshot 5 shows the ball after it has reached the end of the line and has started moving to the left. The last screenshot shows the `JPanel` when the user has pressed the DOWN arrow key and the ball has stopped moving.



The following numbered statements are part of the code. Place the *correct number* in the *correct place* in the `AJPanel` definition on the next page so that the `AJPanel` executes as described above.

```
1.   addKeyListener(this);
2.   int code = e.getKeyCode();
3.   moveAmount = SIZE;
4.   requestFocusInWindow();
5.   g.fillOval(ball.x, ball.y, ball.width, ball.height);
6.   repaint();
7.   ball = new Rectangle(LINE_START_X-SIZE/2,
                                  LINE_Y - SIZE/2, SIZE, SIZE);
8.   addMouseListener(this);
9.   ball.x = ball.x + moveAmount;
10.  moveAmount = -SIZE;
11.  if (code == KeyEvent.VK_UP) {
          t.start();
     } else if (code == KeyEvent.VK_DOWN) {
          t.stop();
     }
12.  t = new Timer(200, this);
13.  g.drawLine (LINE_START_X, LINE_Y, LINE_END_X, LINE_Y);
```

Question/Answer Sheet ID: .....

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

public class AJPanel extends JPanel implements ActionListener,
                                        MouseListener, KeyListener {

     private static final int SIZE = 40;
     private static final int LINE_Y = 70;
     private static final int LINE_START_X = 50;
     private static final int LINE_END_X = LINE_START_X + 10*SIZE;

     private Timer t;
     private Rectangle ball;
     private int moveAmount;

     public AJPanel() {
          moveAmount = SIZE;

          8, 1, 12, 7



     }

     public void keyPressed(KeyEvent e) {

          2
          11

     }

     public void actionPerformed(ActionEvent e) {
          9

          if(ball.x+SIZE >= LINE_END_X) {
               10

          } else if(ball.x <= LINE_START_X) {
               3

          }
          6

     }

     public void paintComponent(Graphics g) {
          super.paintComponent(g);

          5, 13
     }

     public void mousePressed(MouseEvent e) {
          4

     }
```

Question/Answer Sheet ID: .....

```
    public void keyReleased(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {}
    public void mouseClicked(MouseEvent e) {}
    public void mouseEntered(MouseEvent e) {}
    public void mouseExited(MouseEvent e) {}
    public void mouseReleased(MouseEvent e) {}
}
```

(13 marks)

Question/Answer Sheet ID: .....

Question 8 (10 marks)

For this question you need to write the `sameWordLengths()` method. This method is passed two parameters, both arrays of Strings. The method returns `true` if both arrays have corresponding String elements of the same length (i.e. the Strings in position 0 of both arrays have the same length, the Strings in position 1 of both arrays have the same length, etc.) In all other cases the method should return `false`.

Consider the code segment below:

```
String[] words1 = { "hip", "hoop", "he" };
String[] words2 = { "hip", "list", "am" };
String[] words3 = { "sip", "hop", "snip" };
String[] words4 = { "lop", "mope", "in" };

System.out.println("1. " + sameWordLengths(words1,words2));
System.out.println("2. " + sameWordLengths(words1,words3));
System.out.println("3. " + sameWordLengths(words1,words4));
System.out.println("4. " + sameWordLengths(words2,words3));
```

If you have defined the `sameWordLengths()` method correctly, the output from the above code segment should be:

```
1.  true
2.  false
3.  true
4.  false
```

You can assume that both arrays passed to the `sameWordLengths()` method have the same number of elements and no element in the arrays is `null`.

Define the `sameWordLengths()` method in the space provided below:

```
private  boolean  sameWordLengths ( String[] words1,
                                    String[] words2 ){


for (int i = 0; i < words1.length; i++) {
     if (words1[i].length() != words2[i].length()) {
          return false;
     }
}

return true;









}
```

(10 marks)

Question 9 (13 marks)

Question/Answer Sheet ID: .....

Below is the `Coins` class which you need to complete. A program which uses the `Coins` class is shown on the next page. The `Coins` class represents a number of ten cent, twenty cent and fifty cent coins.

```
public class Coins {

     private String name;
     private int coins10, coins20, coins50;

     public Coins(String name, int tens, int twenties, int fifties){
          coins10 = tens;
          coins20 = twenties;
          coins50 = fifties;
          this.name = name;
     }

     public String toString() {
          int numCoins = coins10 + coins20 + coins50;
          String coinStr = "";
          coinStr = coinStr + name + ": " + numCoins + " coins";
          coinStr = coinStr + " (ten cents: " + coins10 + ",
                                        twenty cents: " + coins20;
          coinStr = coinStr + ", fifty cents: " + coins50 + ")";
          return coinStr;
     }

     public int getNumCoins() {
          return coins10 + coins20 + coins50;
     }

     public String getName() {
          return name;
     }

     public void addToCoins(int tens, int twenties, int fifties) {

          coins10 = coins10 + tens;
          coins20 = coins20 + twenties;
          coins50 = coins50 + fifties;

     }

     public double getTotalValue() {

          double value = coins10*10 + coins20*20 + coins50*50;
          value = value / 100.0;
          return value;

     }


     public boolean hasMoreCoins(Coins other) {
```

CONTINUED

Question/Answer Sheet ID: .....

```
        if (getNumCoins() > other.getNumCoins()) {
            return true;
        }
        return false;
    }
}
```

The code shown below uses the `Coins` class.

```
Coins amyCoins = new Coins("Amy", 4, 3, 5);
Coins tomCoins = new Coins("Tom", 0, 1, 8);

System.out.println("1. " + amyCoins.toString());
System.out.println("2. " + tomCoins.toString());

if (amyCoins.hasMoreCoins(tomCoins)) {
    System.out.println("3. " + amyCoins.getName()
                                        + " has more coins");
} else {
    System.out.println("3. " + tomCoins.getName()
                                        + " has more coins");
}

amyCoins.addToCoins(0, 0, 3);
tomCoins.addToCoins(2, 0, 0);
System.out.println("4. " + amyCoins.toString());
System.out.println("5. " + tomCoins.toString());

System.out.println( "6. Amy now has $"
                                    + amyCoins.getTotalValue());
System.out.println( "7. Tom now has $"
                                    + tomCoins.getTotalValue());
System.out.println( "8. Amy now has "
                          + amyCoins.getNumCoins() + " coins");
System.out.println( "9. Tom now has "
                          + tomCoins.getNumCoins() + " coins");
```

Complete the `Coins` class definition above so that the code above compiles and gives the output shown below:

```
1. Amy: 12 coins: (ten cents: 4, twenty cents: 3, fifty cents: 5)
2. Tom: 9 coins: (ten cents: 0, twenty cents: 1, fifty cents: 8)
3. Amy has more coins
4. Amy: 15 coins: (ten cents: 4, twenty cents: 3, fifty cents: 8)
5. Tom: 11 coins: (ten cents: 2, twenty cents: 1, fifty cents: 8)
6. Amy now has $5.0
7. Tom now has $4.4
8. Amy now has 15 coins
9. Tom now has 11 coins
```

(13 marks)

Question/Answer Sheet ID: .....

Question 10 (6 marks)

The following method, `isPrime()`, tests whether or not the value of the parameter is a prime number (a prime number is an integer greater than 1 whose only divisors are 1 and itself).

In the code below, the relational operators are missing. Fill in the blanks in the code with the correct relational operators.

```
private boolean isPrime(int value) {

     int i;

     i = 2;

     while ( (i  <  value) && (value % i  !=  0) ) {

          i++;
     }

     if (i  ==  value) {

          return true;
     } else {
          return false;
     }

}
```

(6 marks)

If you complete the code above correctly, the following statements:

```
System.out.println(isPrime(7));
System.out.println(isPrime(15));
System.out.println(isPrime(29));
```

should produce the output:

```
true
false
true
```

Question/Answer Sheet ID: .....

OVERFLOW PAGE

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

Question/Answer Sheet ID: .....

OVERFLOW PAGE

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

Question/Answer Sheet ID: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may use this page for rough working)

Question/Answer Sheet ID: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may use this page for rough working)