THE UNIVERSITY OF AUCKLAND

SECOND 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** (/13)	**Q7** (/8)	**TOTAL**
Q2 (/9)	**Q5** (/17)	**Q8** (/10)	
Q3 (/8)	**Q6** (/5)	**Q9** (/10)	**(/100)**

Question/Answer Sheet

ID: .....

Question 1 (20 marks)

What is displayed by each of the following pieces of Java program?

a)

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

```
"x"
```

(2 marks)

b)

```
int value = Math.max(1, Math.max(2, 3));
System.out.println("Max is: " + value);
```

```
Max is: 3
```

(2 marks)

c)

```
int value = 13;
System.out.println(6 < value && value < 20);
```

```
true
```

(2 marks)

d)

```
String name = "frantic";
int number = name.indexOf('i');
System.out.println(name.substring(2, number));
```

```
ant
```

Question/Answer Sheet ID: .....

(2 marks)

e)

```
double d = 23.7;
double e = 5.2;
System.out.println((int)(Math.round(d + e)));
```

29

(2 marks)

f)

```
int i = 5;
int[] numbers = { 4, 2, -7, 5, 1, 6, 3 };
System.out.println(numbers[i] + numbers[i+1]);
```

9

(2 marks)

g)

```
String name = "facile";
System.out.println(name.substring(1, 3) +
                                    name.charAt(5));
```

ace

(2 marks)

Question/Answer Sheet ID: .....

h)

```
int x = 8;
int y = 3;
System.out.println(x/y + " * " + y + " + " + x%y +
                                                " == "  + x);
```

```
2 * 3 + 2 == 8
```

(2 marks)

i)

```
int[] numbers = { 2, 3, 4, 5, 6, 7, 8 };
int value, i;
i = 3;
value = Math.max(numbers[i], numbers[i-1]);
System.out.println(value);
```

```
5
```

(2 marks)

j)

```
String word;
String[] s = {"cat", "dog", "snake", "ocelot", "rabbit" };

word = s[4].charAt(0) + s[0].substring(1);
System.out.println(word);
```

```
rat
```

(2 marks)

Question/Answer Sheet

ID: .....

Question 2 (9 marks)

Complete each of the methods below as specified in the comment preceding each method.

a)

```
//method which returns the largest of three numbers

private int largest(int x1, int x2, int x3) {


  return Math.max(x1, Math.max(x2, x3) );




}
```

(3 marks)

b)

```
//method which returns a boolean indicating whether one
//String starts with the same character as another String

//You can assume that both s1 and s2 contain at
//least one character

private boolean sameFirstLetter(String s1, String s2) {


 return s1.charAt(0) == s2.charAt(0);









}
```

(3 marks)

Question/Answer Sheet ID: .....

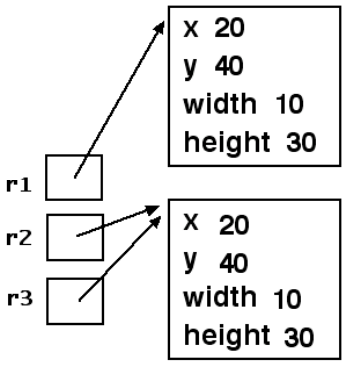
c)

```
//method which displays an array of ints on one
//line with a space after each int

private void displayArray(int[] numbers) {


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

     System.out.println();


}
```

(3 marks)

Question 3 (8 marks)

a) Three `Rectangle` objects can be represented in memory as follows:



Write Java code which creates the three `Rectangle` objects shown above:

```
Rectangle r1, r2, r3;

     r1 = new Rectangle(20, 40, 10, 30);
     r2 = new Rectangle(20, 40, 10, 30);

     r3 = r2;
```

(3 marks)

b) Complete the output when the following code is executed.

Question/Answer Sheet ID: .....

```
Point pt1, pt2, pt3;

pt1 = new Point(60, 20);
pt2 = new Point(pt1);
pt3 = pt2;
System.out.println("1. " + (pt1 == pt2) );

pt1.move(10, 20);
System.out.println("2. " + pt1.x + "," + pt1.y);

pt1.translate(10, 20);
System.out.println("3. " + pt1.x + "," + pt1.y);

pt3.x = pt2.y + 5;
pt2.y = pt1.x + 1;
System.out.println("4. " + pt3.x + "," + pt3.y);
System.out.println("5. " + pt2.x + "," + pt2.y);
```

1. false

2. 10,20

3. 20,40

4. 25,21

5. 25,21

(5 marks)

Question/Answer Sheet

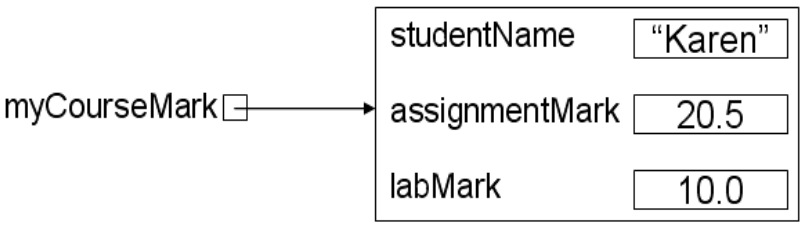
ID: .....

Question 4 (13 marks)

The `CourseMark` class is defined as follows:

```
public class CourseMark {
    private String studentName;       //student name
    private double assignmentMark;    //assignment mark
    private double labMark;           //lab mark

    public CourseMark(String studentName, double labMark,
                                      double assignmentMark) {

        this.studentName = studentName;
        this.assignmentMark = assignmentMark;
        this.labMark = labMark;
    }

    public void setAssignmentMark(double assignmentMark) {
        this.assignmentMark = assignmentMark;
     }

    public double calculateTotal(){
        return assignmentMark + labMark;
    }

    public String toString() {
        return "Name: " + studentName +
                  "\n assignments: " + assignmentMark +
                  "\n labs: " + labMark +
                  "\n coursework total: " +
                  calculateTotal();
    }
}
```

a) Write a Java statement (or statements) which creates an instance of the `CourseMark` class, called "`myCourseMark`", so that the instance has the state visualised as:




```
CourseMark myCourseMark = new CourseMark("Karen",10.0,
                                                    20.5);
```

Question/Answer Sheet ID: .....

(3 marks)

b) Based on your answer in part (a) above, what is the output when the following code is executed?

```
myCourseMark.setAssignmentMark(21.0);
System.out.println(myCourseMark);
```

```
Name: Karen
 assignments: 21.0
 labs: 10.0
 coursework total: 31.0
```

(5 marks)

c) Assume two `CourseMark` objects are equal in value if their coursework totals (i.e. the sum of the assignment mark and the lab mark) are equal. Define the `equals()` method for the `CourseMark` class which returns a `boolean` indicating whether the current `CourseMark` object equals another `CourseMark` object in value.

```
public boolean equals( CourseMark other ) {
 return other.calculateTotal() == calculateTotal();


   OR

 return Math.abs(other.calculateTotal() -
                                    calculateTotal())<0.001;
}
```

(5 marks)

Question/Answer Sheet

ID: .....

Question 5 (17 marks)

The `Music` class is defined as follows:

```
public class Music {
    private String singer;  //name of the singer
    private String song;    //name of the song
    private String type;    //type of the music,
                            //e.g. pop, classical, jazz

    public Music(String singer, String song, String type) {
        this.singer = singer;
        this.song = song;
        this.type = type;
    }

    public String getSinger() {
        return singer;
    }

    public String getSong() {
        return song;
    }

    public String getType() {
        return type;
    }

    public String toString() {
        return singer + ": " + song + " (" + type +")";
    }
}
```

a) Write a Java statement which declares an array of `Music` objects. Use the identifier `myMusicCollection` for the array variable.

```
Music[] myMusicCollection;
```

(2 marks)

Question/Answer Sheet ID: .....

b) Write a Java statement which constructs the `myMusicCollection` array (that you declared in part (a) above) to be large enough to store exactly 500 `Music` objects.

```
myMusicCollection = new Music[500];
```

(2 marks)

c) Write a Java statement which stores the object, `myFavourite`, in index position 0 of the `myMusicCollection` array. The `myFavourite` variable has been declared and initialised for you.

```
Music myFavourite = new Music("Andrea Bocelli",
                        "Besame Mucho", "pop");

myMusicCollection[0] = myFavourite;
```

(2 marks)

For parts (d) to (f) below, assume that the `myMusicCollection` array has been created and contains some `Music` objects.

d) Write a Java statement which assigns the `Music` object in index position 2 of the `myMusicCollection` array to a `Music` variable, `famousMusic`.

```
Music famousMusic = myMusicCollection[2];
```

(3 marks)

e) Write a Java statement which obtains the singer of the `Music` object in index position 3 of the `myMusicCollection` array, and stores it in a `String` variable, `famousSinger`.

```
String famousSinger = myMusicCollection[3].getSinger();
```

(3 marks)

Question/Answer Sheet ID: .....

f) Complete the `countNumberClassical()` method below which processes the `myMusicCollection` array and returns the number of elements of the `myMusicCollection` array which are classical music (i.e. the `type` of the `Music` object is "classical").

Note: the number of actual `Music` objects in the `myMusicCollection` array is given by the parameter, `numberOfRecords`.

```
private int countNumberClassical(Music[] myMusicCollection,
                                 int numberOfRecords) {

   int numberClassical = 0;


   for (int i=0; i<numberOfRecords; i++) {

       if (myMusicCollection[i].getType().equals(
                                              "classical")) {
            numberClassical++;

       }
   }









   return numberClassical;
}
```

(5 marks)

Question/Answer Sheet

ID: .....

Question 6 (5 marks)

Complete the `getNumberContaining()` method which has two parameters, an array of `Rectangle` objects and a `Point` object. The method returns the number of `Rectangles` in the `rects` array which contain the `Point`, `dot`.

```
private int getNumberContaining(Rectangle[]rects, Point dot) {

     int containingDot = 0;

     for (int i=0; i<rects.length; i++) {
          if (rects[i].contains(dot)) {
               containingDot++;
          }

     }
     return containingDot;









}
```

(5 marks)

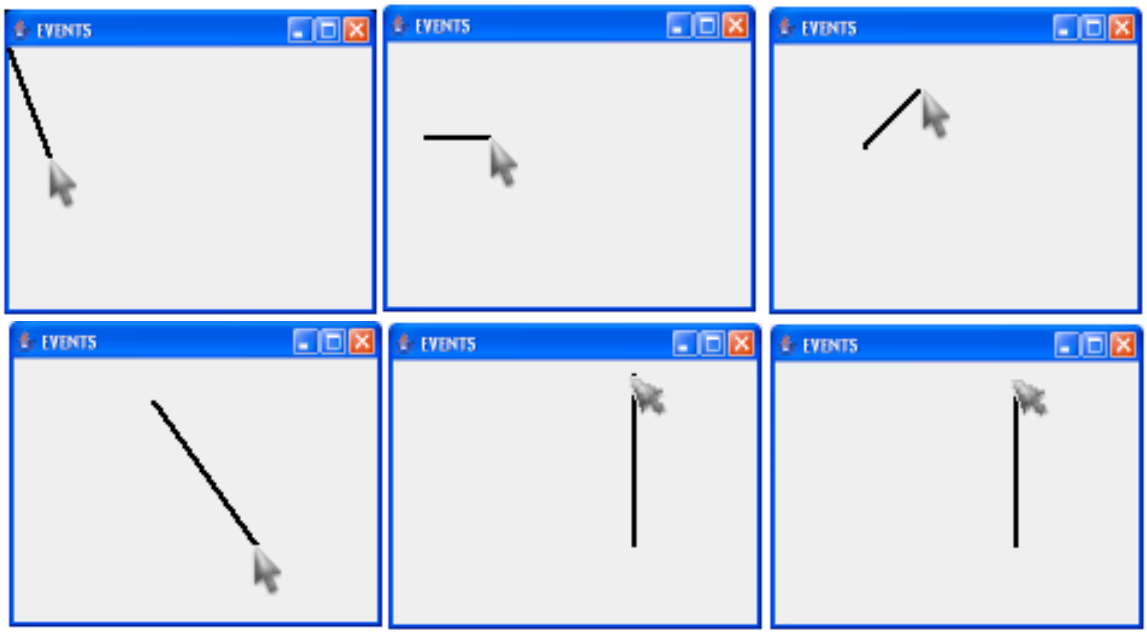
Question/Answer Sheet

ID: .....

Question 7 (8 marks)

The following `JPanel` responds to `MouseEvents`.

Initially the `JPanel` is empty. When the user presses the mouse for the first time, a line is drawn from (0,0) to the position where the user pressed the mouse. From then on, each time the user presses the mouse, a line is drawn from the previous mouse press position to the current mouse press position. The following screen shots show an example of the `JPanel` behaviour for the first six user mouse presses.



The following 14 statements are part of the code. Place each number in the correct place in the AJPanel definition on the next page so the JPanel executes as described above.

```
1.    MouseEvent e
2.    implements MouseListener
3.    repaint();
4.    currentPos = new Point(e.getX(), e.getY());
5.    addMouseListener(this);
6.    Graphics g
7.    public void mouseClicked(MouseEvent e) {}
8.    currentPos = new Point(START_POINT);
9.    extends JPanel
10.   import java.awt.event.*;
11.   previousPos = new Point(START_POINT);
12.   previousPos = currentPos;
13.   g.drawLine(previousPos.x, previousPos.y,
                 currentPos.x, currentPos.y);
14.   private Point currentPos;
```

Question/Answer Sheet ID: .....

```
import java.awt.*;
import javax.swing.*;
 10
public class AJPanel     9              2                    {

  public static final Point START_POINT = new Point(0, 0);
  private Point previousPos;
  14

  public AJPanel() {
     8
     11
     5
  }

  public void paintComponent(      6                    ) {
     super.paintComponent( g );
     13

  }

  public void mousePressed(        1           ) {
     12
     4
     3
  }
  7
  public void mouseReleased(MouseEvent e){}
  public void mouseEntered(MouseEvente){}
  public void mouseExited(MouseEvent e){}

 }
```

(8 marks)

Question/Answer Sheet

ID: .....

Question 8 (10 marks)

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

`minutesL` – a `JLabel` displaying the String, "MINUTES",
`minutesT` – a `JTextField` which contains the time taken for a particular service in minutes as a whole number,
`costB` – a `JButton` displaying the String, "COST OF SERVICE",
`costT` – a `JTextField` which displays, in whole dollars, the cost of the service.

Below is a screenshot of the `JPanel` when it is first displayed. The `minutesT` and the `costT JTextFields` are empty.

In the `minutesT JTextField`, the user enters a whole number, representing the number of minutes taken for a particular service. When the user presses the "COST OF SERVICE" `JButton`, the cost of the service is displayed, in whole dollars, in the `costT JTextField`. The cost is worked out in the following way:

The cost for any service which takes 10 minutes or less is a flat fee of $10.

The cost of any service which takes longer than 10 minutes is $10 plus $2 per minute for every minute over 10.

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 an example of the `JPanel` in action.

CONTINUED

Question/Answer Sheet ID: .....

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

public class AJPanel extends JPanel implements
                                    ActionListener {

     private JTextField minutesT, costT;
     private JButton costB;

     public AJPanel() {
          JLabel minutesL = new JLabel("MINUTES");

          minutesT = new JTextField(5);
          costT = new JTextField(12);

          costB = new JButton("COST OF SERVICE");

          costB.addActionListener( this );

               add(minutesL);
               add(minutesT);
               add(costB);
               add(costT);
          }

     public void actionPerformed ( ActionEvent e ) {

          int minutes =Integer.parseInt(minutesT.getText());

          int moreThan10 = minutes - 10;
          int cost;

          if (minutes < 10) {
               cost = 10;
          } else {
               cost = 10 + moreThan10 * 2;
          }

          costT.setText("$" + cost);


     }
}
```

(10 marks)

Question/Answer Sheet

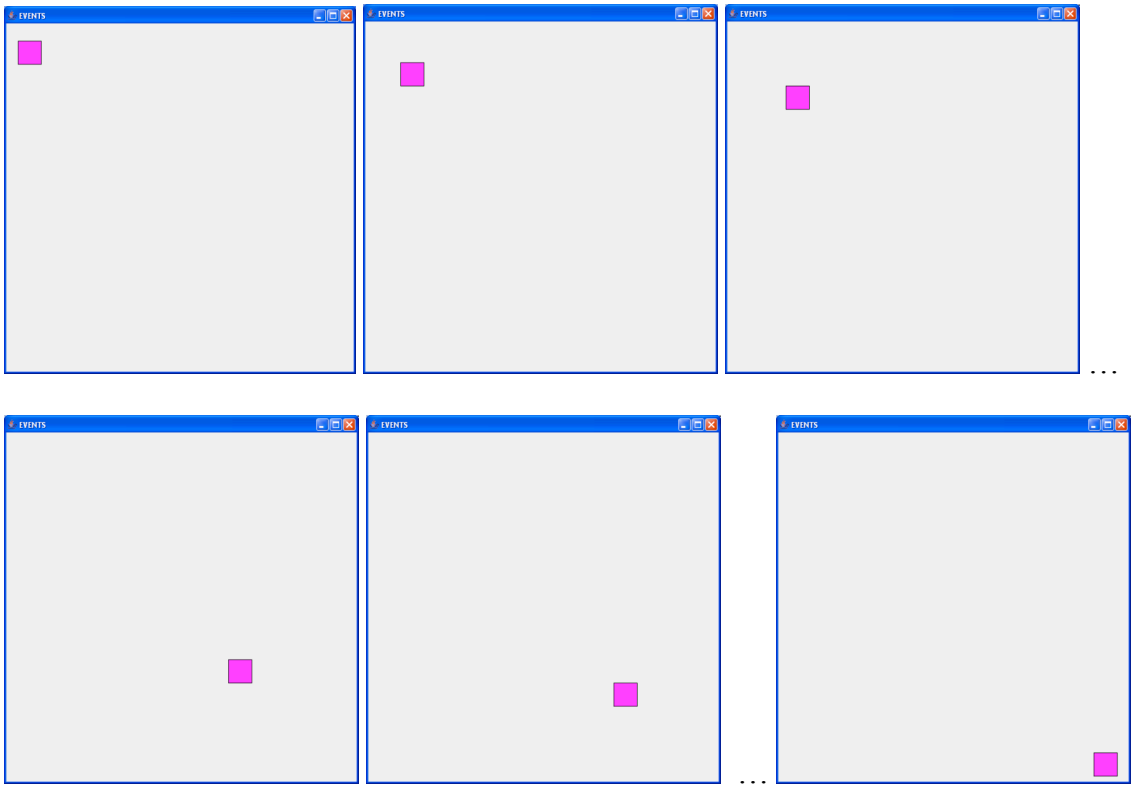
ID: .....

Question 9 (10 marks)

The following `JPanel` uses a `Timer` object. The `Timer` object is created with a delay of 100 milliseconds. The `JPanel` displays a square which moves diagonally down the JPanel. The amount by which the square moves in the horizontal and vertical direction is given by the constant, SIZE. The colour of the filled square is red.

By pressing the UP and DOWN arrow keys, the user controls when the square starts and stops moving. When the user presses the UP arrow key, the `Timer` should start and the square begins to move from its current position. When the user presses the DOWN arrow key, the `Timer` should stop and the square stops in its current position. The square continues moving when the user presses the UP arrow key again, etc.

Below are some screenshots of an example of the `JPanel` in action. The first screenshot shows the `JPanel` when it is first displayed. The square is not moving. The other screenshots show the `JPanel` after the user has pressed the UP arrow key and the square is moving to the bottom right. The last screenshot shows the `JPanel` just before the square moves off the JPanel area.



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 and constants given in the code.

Question/Answer Sheet ID: .....

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

public class AJPanel extends JPanel implements ActionListener,
                                                KeyListener {

    public static final int SIZE = 40;
    public static final Rectangle START_RECT = new
                        Rectangle(20, 30, SIZE, SIZE);

    private Rectangle currentRect;
    private Timer t;

    public AJPanel() {
        addKeyListener( this );
        t = new Timer(300, this );
        currentRect = new Rectangle(START_RECT);
    }

    public void actionPerformed(ActionEvent e) {
        currentRect.x = currentRect.x + SIZE;
        currentRect.y = currentRect.y + SIZE;
        repaint();
    }

    public void keyPressed(KeyEvent e) {
        if ( e.getKeyCode() == KeyEvent.VK_UP ) {
            t.start();
        } else if ( e.getKeyCode() == KeyEvent.VK_DOWN ) {
            t.stop();
        }
    }
```

Question/Answer Sheet ID: .....

```
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.RED);
        g.fillRect(currentRect.x,currentRect.y,SIZE,SIZE);
    }

    public void keyReleased(KeyEvent e) {}
    public void keyTyped(KeyEvent e) {}
}
```

(10 marks)

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)