THE UNIVERSITY OF AUCKLAND

SUMMER SEMESTER, 2011
Campus: City

COMPUTER SCIENCE

Principles of Programming

(Time Allowed: TWO hours)

NOTE:
You must answer **all** questions in this exam.
No calculators are permitted
Answer in the space provided in this booklet.
There is space at the back for answers that overflow the allotted space.

Surname	
Forenames	
Student ID	
Login (UPI)	

Q1 (/8)	**Q4** (/8)	**Q7** (/13)	**Q10** (/7)
Q2 (/9)	**Q5** (/6)	**Q8** (/9)	**Q11** (/7)
Q3 (/8)	**Q6** (/5)	**Q9** (/10)	**Q12** (/10)
		Total	(/100)

ID: .....

Question 1 (8 marks)

a) What output is produced by the following code?

```
int a = 4;
int b = 6;
double c = b % a;

System.out.println(a + a + ", " + b * c + 1);
```

(2 marks)

b) Complete the output of the following code.

```
int a = 4;
int b = a + 6;
double c = 1.0 * b / a;

a = a + (int) (a * c) + 1;
b++;

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

a: b: c:

(2 marks)

ID: .....

c) What output is produced by the following code?

```
int x = 13;
int y = 5;
if (x < 10) {
    if (y != 5) {
       System.out.println("A");
    } else {
       System.out.println("B");
    }
} else if (y > 10) {
    System.out.println("C");
} else {
    if (y != 5) {
       System.out.println("D");
    } else {
       System.out.println("E");
    }
}
```

(2 marks)

d) In the space provided below, assign a value for the variable, `y`, so that the output of the section of code is:

```
AA
```

```
int x = 8;
int y =                ;

if (x < 10 && !(y > 7 || y == x)) {
    if (y % 2 == 0) {
       System.out.println("AA");
    } else {
       System.out.println("BB");
    }
} else if (y > 2) {
    System.out.println("CC");
}
```

(2 marks)

ID: .....

Question 2 (9 marks)

a) What output is produced by the following code?

```
boolean[] a1 = { true, false, false, true };
boolean[] a2 = { false, false, true, true };

for (int i = 0; i < a1.length; i++) {
      System.out.print((a1[i] || !a2[i]) + " ");
}
```

(3 marks)

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

```
String[] s = new String[4];

s[0] = "0";

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

System.out.print("s: ");

for (int i = 1; i < s.length; i++) {
      System.out.print(s[i] + " ");
}
```

```
s:
```

(3 marks)

ID: .....

c) What output is produced by the following code?

```
String[] array = { "3", "to", "One" };

for (int i = 0; i < array.length; i++) {
      array[i] = "-" + array[i].length();
}

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

(3 marks)

ID: .....

Question 3 (8 marks)

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

```
System.out.print("i: ");

for (int i = 15; i > 4; i = i - 3) {
      System.out.print(i + " ");
      i--;
}
```

```
i:
```

(3 marks)

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

```
int toSubtract = 2;
int num = 12;

System.out.print("numbers: ");

while (num >= 0) {
    num = num - toSubtract;
    System.out.print(num + " ");
    toSubtract++;
}
```

```
numbers:
```

(3 marks)

ID: .....

c) The following program does not compile. Give the number of the line of code which causes the compile error. Note: you are not required to write the corrected code.

```
1    public class Program {
2         public void start() {
3              int a = 5;
4              method(a);
5         }
6         private void method(double x) {
7              int b = (int) x;
8              int c = b + a;
9              System.out.println(b + " " + c);
10        }
11   }
```

```
Line number of the code
which causes the compile error:
```

(2 marks)

ID: .....

Question 4 (8 marks)

a) Complete the output when the following start() method is executed.

```
public void start() {
      Point p1 = new Point(5, 9);
      Point p2 = new Point(p1.y, p1.x);

      p1.move(2, 3);
      p2.translate(2, 3);

      System.out.println("1: " + p1.x + ", " + p1.y);
      System.out.println("2: " + p2.x + ", " + p2.y);
}
```

```
1:

2:
```

(2 marks)

b) Complete the following section of code which assigns the centre point of the `Rectangle`, `r`, to the `Point` variable, `midPoint`;

```
Rectangle r = new Rectangle( .... );

int midRectX = [                                ];

int midRectY = [                                ];

Point midPoint = new Point(midRectX, midRectY);
```

(2 marks)

c) Give the output when the following start() method is executed.

```
public void start() {
      Point pt = new Point(5, 9);

      fiddle(pt);

      System.out.println("1: " + pt.x + ", " + pt.y);
}

private void fiddle(Point p1) {
      p1 = new Point(p1.y, p1.x);
      Point p2 = p1;
      p2.y = 6;
      System.out.println("2: " + p1.x + ", " + p1.y);
}
```

(2 marks)

d) Give the output when the following start() method is executed.

```
public void start() {
      Point pt = new Point(3, 7);

      fiddle(pt);

      System.out.println("1: " + pt.x + ", " + pt.y);

}

private void fiddle(Point p1) {
      Point p2 = new Point(p1);
      p2.x = p2.x + 2;
      System.out.println("2: " + p1.x + ", " + p2.x);
}
```

(2 marks)

ID: .....

Question 5 (6 marks)

Give the output when the following program is executed.

```
public class MyProgram {
     public void start() {
          int a = method2('Z');
          System.out.println("1. " + a);

          String s = method1("ABCD", 5);
          System.out.println("2. " + s);
     }

     private String method1(String a, int b) {
          char letter = a.charAt(0);

          if (b > 4) {
               letter = a.charAt(1);
          }

          int d = method2(letter);
          System.out.println("3. " + d);

          return d + a;
     }

     private int method2(char c) {
          if (c == 'A') {
               return 1;
          }

          if (c == 'B') {
               return 2;
          }

          System.out.println("4. " + c);

          return -1;
     }
}
```

ID: .....

```
//Show output here
```

(6 marks)

ID: .....

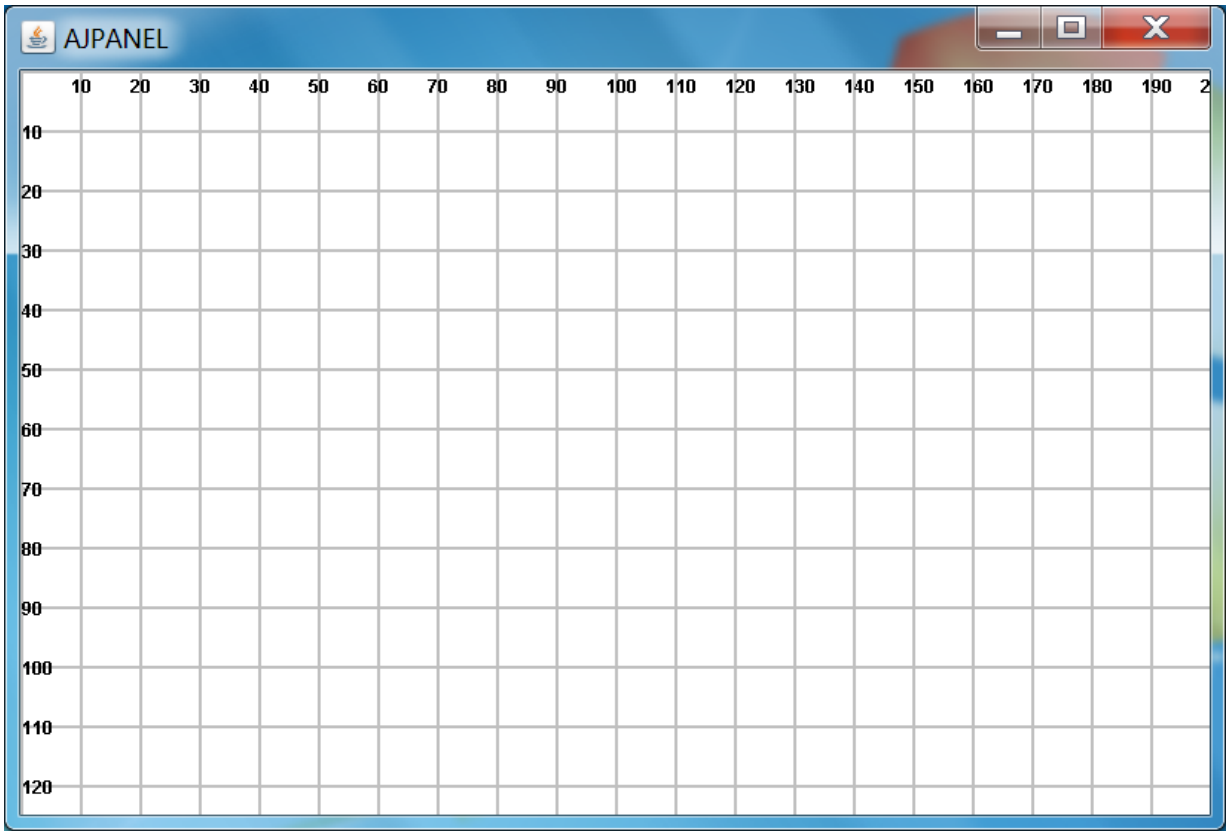
Question 6 (5 marks)

As accurately as possible, show what would be drawn in the window by the following program. Grid lines have been drawn on the window to help you.

The gap between adjacent gridlines is 10 pixels.

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

public class MyJPanel extends JPanel {

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

      drawPattern(g, 20, 30, 10);
   }

   private void drawPattern(Graphics g, int x, int y,
                                                  int size) {
      g.drawString("Boo", x, y);

      for (int i = 0; i < 2; i++) {

          g.drawRect(x, y, size * 2, size);

          x = x + size / 2;

          g.fillOval(x, y, size, size);

          g.drawLine(x + size, y + size, x, y + size * 2);

          x = x + size * 3;
      }

   }
}
```

ID: .....


AJPANEL
10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180 190 2
10
20
30
40
50
60
70
80
90
100
110
120

(5 marks)

ID: .....

Question 7 (13 marks)

All the parts of this question are based on the Car class.
The Car class definition below is incomplete.

```
public class Car {
  public static final int WHITE = 0;
  public static final int RED = 1;
  public static final int SILVER = 2;

  private int colour;
  private int mileage;
  private String make;
  private String licencePlate;

  public Car(...) {
       ...
  }

  public Car(int colour, String make) {
      this.colour = colour;
      this.make = make;
  }
  public ...  getMileage() {
        ...
  }
  public ...  setMileage(...) {
        ...
  }
  public String getColour() {
      if (colour == WHITE) {
         return "White";
      } else if (colour == RED) {
         return "Red";
      } else {
         return "Silver";
      }
  }
  public boolean hasDoneMoreMilesThan(...) {
        ...
  }
  public String toString() {
      return colour + make + licencePlate + mileage;
  }
}
```

ID: .....

a) Define the constructor method which initialises the values of the four instance variables. Below is an example call to the constructor:

```
Car c1 = new Car(Car.RED, "AWESUM", "Ferrari", 5000);
```

```
public _____________ ( ____________________________________
            ____________________________________________ ) {









}
```

(2 marks)

b) Complete the accessor method, `getMileage()`.

```
public ____________________________________ () {


}
```

(2 marks)

c) Complete the mutator method, `setMileage()`.

```
public ____________________________ ( _________________ ) {


}
```

(2 marks)

d) Complete the `hasDoneMoreMilesThan()` method. This method returns `true` if a car has done more miles than another car, `false` otherwise. For example, the following code:

```
Car c1 = new Car(Car.RED, "AWESUM", "Ferrari", 5000);
Car c2 = new Car(Car.SILVER, "OH MY", "Maserati", 400);

if (c1.hasDoneMoreMilesThan(c2)) {
      System.out.println("More miles");
}
```

gives the output:

```
More miles
```

```
public boolean hasDoneMoreMilesThan( ____________ ) {



}
```

(3 marks)

e) Give the output when the following program is executed.

```
Car c1 = new Car(Car.RED, "AWESUM", "Ferrari", 5000);
Car c2 = new Car(Car.WHITE, "Subaru");

System.out.println(c1.getColour());
System.out.println(c1.toString());
System.out.println(c2.toString());
```

(4 marks)

ID: .....

THIS PAGE

HAS BEEN

INTENTIONALLY

LEFT BLANK

ID: .....

Question 8 (9 marks)

All the parts of this question are based on the Shape class.
The `Shape` class definition below is incomplete.

```
public class Shape {
  private int numSides;
  private String name;

  public Shape(String name, int numSides) {
      this.name = name;
      this.numSides = numSides;
  }

  public void print(Shape other) {
      System.out.println(toString());
      System.out.println(equals(other));
  }

  public String toString() {
      return name + " has " + numSides + " sides";
  }

  public boolean equals(Shape other) {
      return this.numSides == other.numSides;
  }
}
```

a) Give the output when the following code is executed.

```
Shape square = new Shape("Square", 4);
Shape diamond = new Shape("Diamond", 4);

square.print(diamond);
```

(3 marks)

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

```
Shape square = new Shape("square", 4);
Shape circle = new Shape("circle", 4);

System.out.println(square.equals(circle));
```

(2 marks)

c) Rewrite the `equals()` method so that the output when the following code is executed:

```
Shape shape1 = new Shape("square", 4);
Shape shape2 = new Shape("SQUARE", 4);
Shape shape3 = new Shape("diamond", 4);

System.out.println(shape1.equals(shape2));
System.out.println(shape1.equals(shape3));
```

is:

```
true
false
```

```
public boolean equals(Shape other) {









}
```

(4 marks)

ID: .....

Question 9 (10 marks)

All the parts of this question are based on the Item and ShoppingList classes.
The `ShoppingList` class definition below is incomplete.

```
public class Item {
  private String name;
  private double unitCost;
  private int quantity;
  public Item(String name, double unitCost, int quantity) {
      this.name = name;
      this.unitCost = unitCost;
      this.quantity = quantity;
  }
  public String getName() { return name; }
  public double getUnitCost() { return unitCost; }
  public int getQuantity() { return quantity; }
}
```

```
public class ShoppingList {
  private Item[] items;
  private int nextPos;
  public ShoppingList(int maxSize) {
      items = new Item[maxSize];
      nextPos = 0;
  }
  public void add(Item item) {
      ...
  }
  public double getTotal() {
      ...
  }
}
```

a) Complete the `add()` method for the `ShoppingList` class so that the specified item is added to the `items` array. **Note:** you may assume that there is always enough space for a new `Item` object to be added to the `items` array.

```
public void add(Item item) {
```

ID: .....

```
}
```

(5 marks)

b) Complete the `getTotal()` method for the `ShoppingList` class so that it returns the total cost of all the `Item` objects currently stored in the `items` array. For example, the following code, executed using the completed `ShoppingList` class,

```
Item chicken = new Item("Chicken", 10.50, 5);
Item chocs = new Item("Chocolates", 2, 10);

ShoppingList list = new ShoppingList(10);

list.add(chicken);
list.add(chocs);

System.out.println("Total cost is $" + list.getTotal());
```

produces the output:

```
Total cost is $72.5
```

```
public double getTotal() {









}
```

(5 marks)

ID: .....

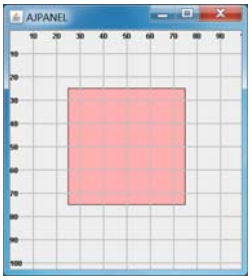
Question 10 (7 marks)

The following `JPanel` responds to `MouseEvents`.

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
public class AJPanel implements MouseListener {
  private final Rectangle AREA = new Rectangle(25, 25, 50, 50);
  private Point[] positions;
  private int upTo;
  public AJPanel() {
      addMouseListener(this);
      positions = new Point[20000];  //assume the array is always
                                                          big enough
      upTo = 0;
  }
  public void mousePressed(MouseEvent e) {
      int x = e.getX();
      int y = e.getY();
      boolean isOK = checkPosition(x, y);
      if (isOK) {
            positions[upTo] = new Point(x, y);
            upTo++;
      } else {
            upTo = 0;
      }
      repaint();
  }
  private boolean checkPosition(int x, int y) {
      Point pointToTest = new Point(x, y);
      if (AREA.contains(pointToTest)) {
            return true;
      }
      return false;
  }
  public void paintComponent(Graphics g) {
      super.paintComponent(g);
      g.setColor(Color.PINK); //Draw the pink filled rectangle
      g.fillRect(AREA.x, AREA.y, AREA.width, AREA.height);
      g.setColor(Color.BLACK);
      g.drawRect(AREA.x, AREA.y, AREA.width, AREA.height);
      Point point1, point2;
      for(int i = 0; i < upTo - 1; i++) {
            point1 = positions[i];
            point2 = positions[i + 1];
            g.drawLine(point1.x, point1.y, point2.x, point2.y);
      }
  }
  public void mouseClicked(MouseEvent e) {}
  public void mouseEntered(MouseEvent e) {}
  public void mouseExited(MouseEvent e) {}
  public void mouseReleased(MouseEvent e) {}
}
```

Note: the grid lines have been drawn in the window to help you.

Initially, the `JPanel` displays a filled pink rectangle (see the `JPanel` on the right). This rectangular area is defined by the constant, `AREA`.

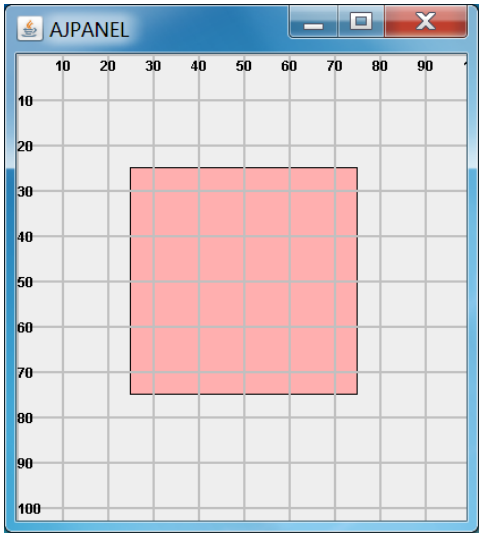


CONTINUED

ID: .....

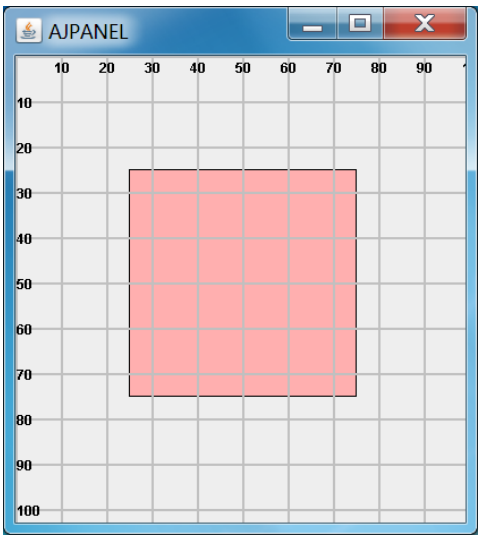
Show the `JPanel` after the user has pressed the mouse three times:

first mouse press occurs at position 40, 30
second mouse press occurs at position 70, 60
third mouse press occurs at position 50, 70



Now show the `JPanel` after the user has pressed the mouse another three times:

fourth mouse press occurs at position 90, 80
fifth mouse press occurs at position 30, 70
sixth mouse press occurs at position 50, 50



(7 marks)

ID: .....

Question 11 (7 marks)

The `JPanel` defined below contains the following components:

`markL`	a `JLabel` displaying the String “Mark”,
`markT`	a `JTextField` where the user will enter a mark,
`outOfL`	a `JLabel` displaying the String “Out Of”,
`outOfT`	a `JTextField` where the user will enter what the mark is out of,
`resultT`	a `JTextField` which will display information to the user,
`checkB`	a `JButton` displaying the String “CHECK”,
`clearB`	a `JButton` displaying the String “CLEAR”.

Below is a screenshot of the `JPanel` when it first appears. Initially the three `JTextFields` are empty.

The user enters an integer value in the `'Mark' JTextField` and an integer value in the `'Out Of' JTextField`. When the user presses the “CHECK” button, the program checks if the mark entered by the user is a pass or not. If the mark is a passing mark (i.e. it is greater than or equal to 50%), the program displays "PASS" in the `resultT JTextField`, otherwise the String "NOT A PASS" is displayed in the `resultT JTextField`. When the user presses the “CLEAR” button, all three `JTextFields` are cleared.

You are required to complete the following `JPanel` definition so that the `JPanel` behaves as described above. You **MUST** use the variables given in the code.

The two screenshots below show the `JPanel` after the user has entered a mark, the value which the mark is out of and then pressed the “CHECK” button.

The last screenshot shows the `JPanel` after the user presses the "CLEAR" button.


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


```
public class AJPanel extends JPanel
```

{

```
    private JTextField markT, outOfT, resultT;
    private JButton checkB, clearB;
```

ID: .....

```
public AJPanel() {
     JLabel markL = new JLabel("Mark");
     JLabel outOfL = new JLabel("Out Of");

     markT = new JTextField(3);
     outOfT = new JTextField(3);
     resultT = new JTextField(9);

     checkB = new JButton("CHECK");
     clearB = new JButton("CLEAR");

     add(markL);
     add(markT);
     add(outOfL);
     add(outOfT);
     add(resultT);
     add(checkB);
     add(clearB);
```

}

```
public void _____ ( _____ e) {
```

ID: .....

```
    }
}
```

(7 marks)

ID: .....

Question 12 (10 marks)

You are required to complete the following `JPanel` which uses a `Timer` object. The `Timer` object is created with a delay of 300 milliseconds and the `Timer` starts as soon as the `JPanel` is created.

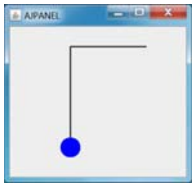
The `JPanel` displays two lines at right angles to each other and a blue ball of size 40 pixels (given by the constant, `SIZE`) which initially is placed with its centre at the bottom of the vertical line. With each tick of the `Timer` the ball moves 40 pixels (given by the constant, `SIZE`) up the vertical line until it reaches the top of the vertical line. Then the ball moves along the horizontal line until it reaches the end of the horizontal line. When the ball reaches the end of the horizontal line it is no longer able to move.

The user controls the `JPanel` using the arrow keys in the following way:

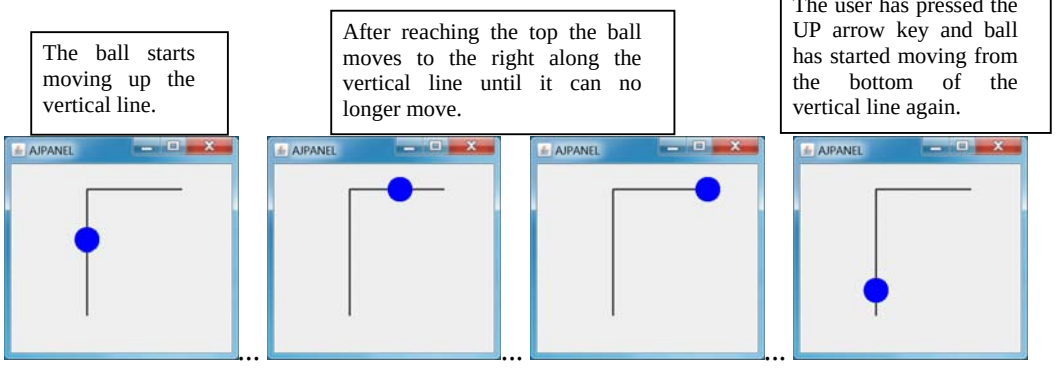
The DOWN arrow key
At any stage the user can stop/restart the `Timer` by pressing the DOWN arrow key, i.e. if the `Timer` is running the Timer will stop, otherwise the `Timer` will start again.

The UP arrow key
Whenever the user presses the UP arrow key, the ball will be reset back to its initial position with its centre at the bottom of the vertical line. If the `Timer` is running then the ball will immediately start moving up the vertical line. If the `Timer` is not running the ball remains stationary until the user presses the DOWN arrow key. When the ball starts moving again it will move up the vertical line.

The screenshot on the right shows the `JPanel` when it is first displayed.



Below are some screenshots of the `JPanel` in action.



Notes:

1. The variables, `centreX` and `centreY`, represent the centre of the blue ball.

2. You MUST **use the variables and constants given in the code.**

```
import javax.swing.*;
```

ID: .....

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

public class AJPanel extends JPanel
                                                                    {
          //The size of the blue ball
private final int  SIZE = 40;
          //The left and right values of the horizontal line
private final int LINE_LEFT = 120;
private final int LINE_RIGHT = LINE_LEFT + 150;
          //The top and bottom values of the vertical line
private final int LINE_TOP = 40;
private final int LINE_BOTTOM = LINE_TOP + 200;
             //The two directions in which the ball travels
public static final int UP = 0;
public static final int RIGHT = 1;

private int centreX, centreY;
private int direction;
private Timer t;

public AJPanel() {
     centreX = LINE_LEFT;
     centreY = LINE_BOTTOM;
     direction = UP;
```

```
}
public void actionPerformed(ActionEvent e) {
```

ID: .....

```
}
public void paintComponent(Graphics g) {
      super.paintComponent(g);
      g.drawLine(LINE_LEFT, LINE_TOP, LINE_RIGHT, LINE_TOP);
      g.drawLine(LINE_LEFT, LINE_TOP, LINE_LEFT, LINE_BOTTOM);
      g.setColor(Color.BLUE);
```

```
}
public void keyPressed(KeyEvent e) {
```

ID: .....

```
  }

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

(10 marks)

ID: .....

OVERFLOW PAGE

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

ID: .....

OVERFLOW PAGE

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

ID: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may detach this page from the answer booklet and use it for rough working)

ID: .....

ROUGH WORKING (WILL NOT BE MARKED)

(You may detach this page from the answer booklet and use it for rough working)