THE UNIVERSITY OF AUCKLAND

SUMMER SEMESTER, 2012
Campus: City

COMPUTER SCIENCE

Principles of Programming

(Time Allowed: TWO hours)

NOTE:
You must answer **all** questions in this test.
No calculators are permitted
Answer in the spaces 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** (/12)	**Q10** (/12)
Q2 (/6)	**Q5** (/10)	**Q8** (/8)	**Q11** (/12)
Q3 (/6)	**Q6** (/10)	**Q9** (/8)	**Total** /100

ID: .....

Question 1 (8 marks)

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

```
System.out.println(3 + 3 * 2 + ", " + 5 * 5 + 2 + 1);
```

(2 marks)

b) Complete the output produced by the following code:

```
int a = 10;
int b = a * 2 + a / 3;
double c = b / a * 2.0;

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

```
a:          b:          c:
```

(2 marks)

c) If the variable, `a`, is defined as follows:

```
int a = (int) (Math.random() * 5) + 3;
```

what is the smallest possible number which could be assigned to the variable, `a` and what is the largest possible number which could be assigned to the variable, `a`?

```
Smallest possible value for a is

Largest possible value for a is
```

(2 marks)

d) Complete the output produced by the following code:

```
String word = "Fantastic";
int pos = word.indexOf("tick");
String result = word.substring(pos + 2);

System.out.println("pos: " + pos + " result: " + result);
```

```
pos:              result:
```

(2 marks)

ID: .....

Question 2 (6 marks)

a) What is the output when the following program is executed?

```
public class Program {

  public void start() {
      printLetters("Higgledy");
  }

  private void printLetters(String letters) {
      String version1 = letters.toLowerCase();
      String version2 = letters.toUpperCase();

      for(int i = 0; i < letters.length(); i++) {
         if (i % 2 == 0) {
              System.out.print(version1.charAt(i));
         } else {
              System.out.print(version2.charAt(i));
         }
      }
  }
}
```

(2 marks)

b) Complete the `for` loop so that the output is exactly:

```
27   24   21   18   15   12   9
```

```
 for(              ;                    ;              ) {
      System.out.print(i + "   ");
}
```

(2 marks)

c) What is the output when the following program is executed?

```
public class Program {

  public void start() {

       int[] numbers = {1, 3, 2, 4, 0, 3, 7, 4, 5, 2, 6, 6};

       int i = 0;
       int count = 3;

       while (count > 0 && i < numbers.length) {

           if (numbers[i] < 3) {
                System.out.print(numbers[i] + " ");
                count--;
           }

           i++;
       }
  }
}
```

(2 marks)

ID: .....

Question 3 (6 marks)

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

```
int a = 4;
int b = 6;
int c = 3;
boolean d = false;

boolean result = a > c && ! (a >= b || d);
System.out.println("result: " + result);
```

```
result:
```

(2 marks)

b) What is the output when the following program is executed?

```
public class Program {
  public void start() {
      lotsOfIfs(12, 6);
  }
  private void lotsOfIfs(int x, int y) {
      if (x < 10 && y > 4) {
          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");
          }
          System.out.print("E");
      }
  }
}
```

(2 marks)

ID: .....

c) What is the output when the following program is executed?

```
public class Program {
  public void start() {
      lotsOfIfs(12, 6);
  }
  private void lotsOfIfs(int x, int y) {
      if (x < 8 || (y > 5 && y != x)) {
          if (y > x && x < 12) {
              System.out.println("A");
          } else {
              System.out.println("B");
          }
      } else if (! (y > 2 || y < 6)) {
          if (x >= 0 && ! (x < 7)) {
              System.out.println("C");
          } else {
              System.out.println("D");
          }
          System.out.println("E");
      }

      if (y + x > 10) {
          System.out.println("F");
      }
  }
}
```

(2 marks)

ID: .....

Question 4 (8 marks)

a) Complete the `getCost()` method which is passed an `int` parameter, `minutes`. First the method calculates the number of hours which fully contain the minutes (e.g., 0 minutes is 0 hours, 1-60 minutes is 1 hour, 61-120 minutes is 2 hours, 121-180 minutes is 3 hours, etc.). Each hour costs $3, but, if the final cost is greater than $10, then the maximum cost is $10. The method returns the final cost. For example, the following code:

```
int cost = getCost(43);
System.out.println("43 minutes cost $" + cost);

cost = getCost(120);
System.out.println("120 minutes cost $" + cost);

cost = getCost(600);
System.out.println("600 minutes cost $" + cost);
```

produces the following output:

```
43 minutes cost $3
120 minutes cost $6
600 minutes cost $10
```

```
private int getCost(int minutes) {
    final int COST_PER_HOUR = 3;









}
```

(4 marks)

ID: .....

b) Complete the `startEndAreTheSame()` method which is passed a String parameter, `word`. The method returns `true` if the first two letters of the String parameter are the same as the last two letters. Otherwise, the method returns `false`. For example, the following code:

```
boolean result = startEndAreTheSame("decide");
System.out.println("decide - " + result);

result = startEndAreTheSame("decided");
System.out.println("decided - " + result);

result = startEndAreTheSame("go");
System.out.println("go - " + result);
```

produces the following output:

```
decide - true
decided - false
go – true
```

Note: you may assume that the String parameter always contains 2 or more letters.

```
private boolean startEndAreTheSame(String word) {









}
```

(4 marks)

ID: .....

Question 5 (10 marks)

a) Complete the following method which takes a `Rectangle r` as a parameter and returns a larger one with the same top left position, but width and height both twice that of `r`.

```
private Rectangle enlarge(Rectangle r) {
    Rectangle r2 =
```

(2 marks)

```
    return r2;
}
```

b) Complete the definition of the `Rectangle r2` so that the following code prints the letter A.

```
private void testRectMethod() {
    Rectangle r1 = new Rectangle(30, 50, 100, 40);
    r1.height = r1.height * 3;
    Rectangle r2 =
```

(2 marks)

```
    if (isSkinnier(r2, r1)) {
      System.out.println("A");
    } else {
      System.out.println("B");
    }
}
private boolean isSkinnier(Rectangle r1, Rectangle r2) {
    if (r1.width < r2.width && r1.height > r2.height) {
      return true;
    }
    return false;
}
```

ID: .....

c) Give the output produced when the following code is executed.

```
Rectangle r1 = new Rectangle(20, 50, 40, 40);
Rectangle r2 = new Rectangle(80, 10, 50, 60);
Point p = new Point(90, 65);

r2 = r1;
r2.x = r2.x + 15;

if (r1.contains(p)) {
     System.out.print("A ");
} else if (r2.intersects(r1)) {
     System.out.print("B ");
}

if (r1.equals(r2)) {
     System.out.print("C ");
} else {
     System.out.print("D ");
}
```

(2 marks)

d) Complete the `largestRectangleArea()` method. This method takes an array of rectangles, `rectArray`, as a parameter and returns the area of the largest rectangle in the array. You may assume the array is full and contains no `null` elements.

```
private int largestRectangleArea(                          ){
     int maxArea = 0;









     return maxArea;
}
```

(4 marks)

ID: .....

Question 6 (10 marks)

a) Complete the `moveDown()` method which is passed two parameters: an `int[]` array, `values`, and the index of the first element of the array to be replaced. The method moves all the elements from the position, `start`, to the end of the array down one position. (Note that the element at position, `start`, will be lost.) The last element of the array is assigned the value 0. For example, the following code:

```
int[] nums = { 4, 6, 2, 5, 8, 9, 0, 6, 1, 8, 7, 2 };
printArray(nums);

System.out.println("Move elements from position 3");
moveDown(nums, 3);
printArray(nums);

System.out.println("Move elements from position 5");
moveDown(nums, 5);
printArray(nums);

System.out.println("Move elements from position 0");
moveDown(nums, 0);
printArray(nums);
```

produces the following output:

```
4   6   2   5   8   9   0   6   1   8   7   2
Move elements from position 3
4   6   2   8   9   0   6   1   8   7   2   0
Move elements from position 5
4   6   2   8   9   6   1   8   7   2   0   0
Move elements from position 0
6   2   8   9   6   1   8   7   2   0   0   0
```

```
private void moveDown(int[] values, int start) {









}
```

(5 marks)

ID: .....

b) What is the output when the following program is executed?

```
public class Program {
  public void start() {
      int[] nums1 = { 1, 2, 3, 4, 5, 6, 7, 8, 9 };

      int[] nums2 = getEvenOddsArray(nums1);

      for (int i = 0; i < nums2.length; i++) {
         System.out.print(nums2[i] + " ");
      }
  }

  private int[] getEvenOddsArray(int[] a) {
      int start = 0;
      int end = a.length - 1;
      int[] b = new int[a.length];

      for (int i = 0; i < a.length; i++) {
         if (a[i] % 2 == 0) {
              b[start] = a[i];
              start++;
         } else {
              b[end] = a[i];
              end--;
         }
      }

      return b;
  }
}
```

(5 marks)

ID: .....

Question 7 (12 marks)

The `CarWash` class definition below is incomplete. Code which uses the `CarWash` class is shown on the next page.

```
public class CarWash {
    private int option;
    private boolean withTyreShine;

    public CarWash(int optionSelected, boolean withShine) {
        option = optionSelected;
        this.withTyreShine = withShine;
    }

    public          getOption() {


    }
```

(2 marks)

```
    public void setOption(int optionSelected) {
        option = optionSelected;
    }

    public         setWithTyreShine(                    ) {


    }
```

(2 marks)

```
    public boolean getWithTyreShine() {
        return withTyreShine;
    }
    public int getTotalCost() {
        int cost = 0;

        if (option == 1) {
            cost = 35;
        } else {
            cost = 55;
        }
        if (withTyreShine) {
            cost = cost + 12;
        }

        return cost;
    }

    public          isMoreExpensive(                    ) {









    }
```

(3 marks)

CONTINUED

ID: .....

```
    public String toString() {
        String info = getWashDescription();
        info = info + " $" + getTotalCost();
        return info;
    }
    private String getWashDescription() {
        String serviceInfo = "Clean outside";
        if (option == 2) {
            serviceInfo = "Clean inside & outside";
        }
        if (withTyreShine) {
            serviceInfo = serviceInfo + " + tyre shine";
        }
        return serviceInfo;
    }
}
```

a) In the space provided on the previous page, complete the `getOption()` accessor method.

b) In the space provided on the previous page, complete the `setWithTyreShine()` mutator method.

c) In the space provided on the previous page, complete the `isMoreExpensive()` method. This method should `true` if the `CarWash` object costs more than the parameter `CarWash` object, otherwise the method returns `false`.

d) Assuming the `CarWash` class is defined as specified, give the output when the following code is executed:

```
CarWash wash1 = new CarWash(1, false);
CarWash wash2 = new CarWash(2, true);
System.out.println("1. " + wash1.toString());
System.out.println("2. " + wash2.toString());
wash2.setOption(1);
wash1.setWithTyreShine(true);
wash2.setWithTyreShine(false);
if (wash1.isMoreExpensive(wash2)) {
    System.out.println("3. ");
} else {
    System.out.println("4. ");
}
System.out.println("5. " + wash1.toString());
System.out.println("6. " + wash2.toString());
```

ID: .....

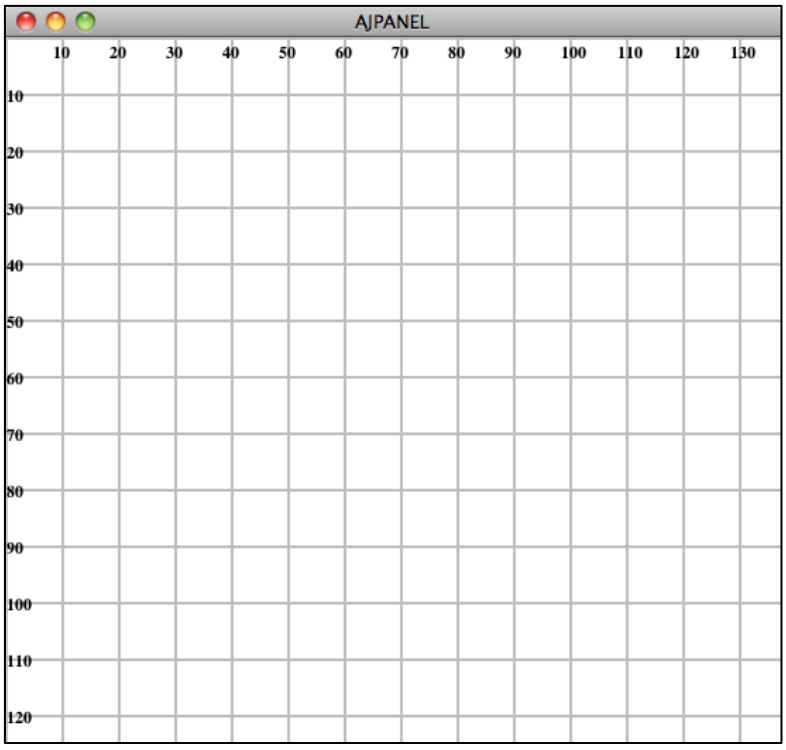
(5 marks)

ID: .....

Question 8 (8 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 AJPanel extends JPanel {

  public void paintComponent(Graphics g){
      super.paintComponent(g);
      drawLogo(g, 30, 40, 10);
  }

  public void drawLogo(Graphics g, int left, int top,
                                                    int size) {

      g.drawRect(left, top, 8 * size, 4 * size);

      g.drawLine(left - size, top - size, left, top);

      g.drawString("Blobs!", left, top + 4 * size);

      for(int i = 0; i < 2; i++) {
          g.fillOval(left, top, size, size);
          left = left + size;
          size = size * 2;
      }
  }
}
```



(8 marks)

CONTINUED

ID: .....

Question 9 (8 marks)

The following JPanel responds to MouseEvents.

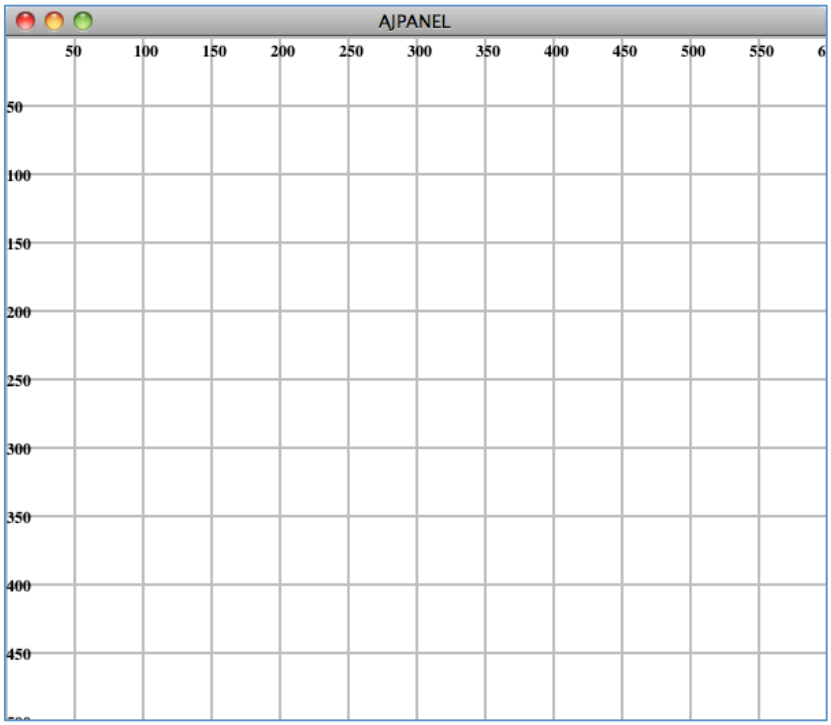
```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class AJPanel extends JPanel implements MouseListener {
    private final int DEFAULT_SIZE = 100;
    private Rectangle[] squares;
    private int number;

    public AJPanel() {
         addMouseListener(this);
                                        //Assume this will be big enough
         squares = new Rectangle[1000];
         number = 0;
    }
    public void mousePressed(MouseEvent e) {
         Point pressPt = e.getPoint();
         boolean newSquareNeeded = true;

         for(int i = 0; i < number; i++) {
             if (squares[i].contains(pressPt)) {
                 squares[i].width = squares[i].width + 100;
                 squares[i].height = squares[i].height + 100;
                 newSquareNeeded = false;
             }
         }

         if(newSquareNeeded) {
             squares[number] = new Rectangle(pressPt.x,
                               pressPt.y, DEFAULT_SIZE, DEFAULT_SIZE);
             number++;
         }

         repaint();
    }

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

         for (int i = 0; i < number; i++) {
             g.drawRect(squares[i].x, squares[i].y,
                                squares[i].width, squares[i].height);
         }
    }

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

ID: .....

Note: the grid lines have been drawn in the window to help you. The gap between adjacent gridlines is 50 pixels.
Initially, the `JPanel` is blank until the user clicks within it.

a) Show the state of the `JPanel` after the user has pressed the mouse three times at the following positions:

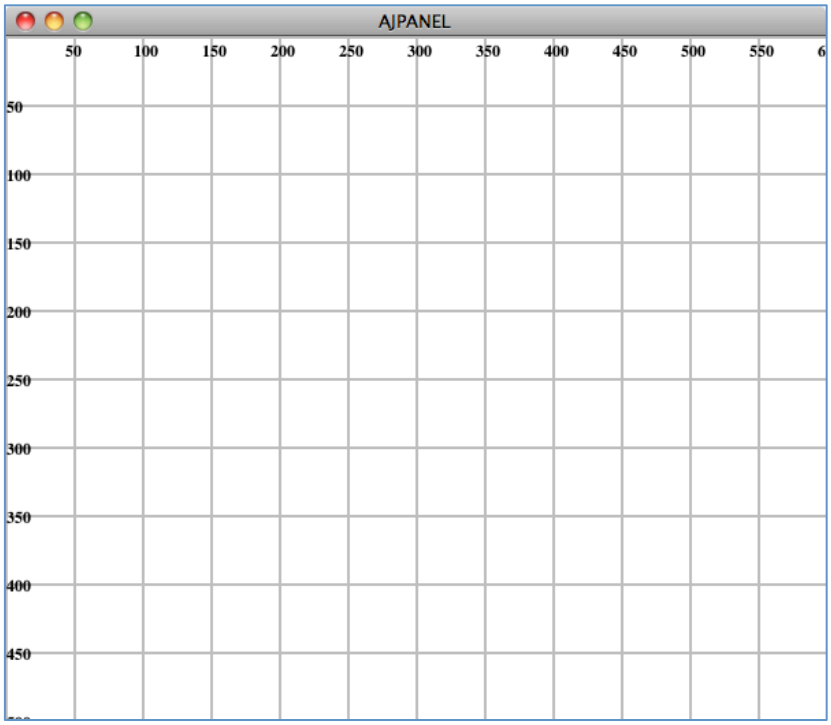
position 150, 100,
position 300, 250,
position 350, 300



(4 marks)

b) Show the state of the `JPanel` after the user has pressed the mouse two more times at the following positions:

position 200, 150,
position 125, 250



(4 marks)

CONTINUED

ID: .....

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK

ID: .....

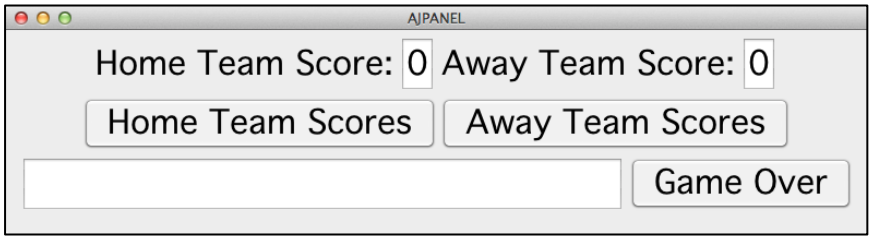
Question 10 (12 marks)

The `JPanel` defined below acts as a scoreboard and contains the following components:

`homeT, awayT`	two `JTextFields` which will display the home team score and the away team score.
`homeB, awayB`	two `JButtons` which will increment the home and away scores, respectively.
`messageT`	a `JTextField` which is initially empty and will display the game result.
`gameOverB`	a `JButton` which says "Game Over".

As well as these components, the `JPanel` has two integer instance variables, `homeScore` and `awayScore`, which store the current scores of the home and away teams, respectively.

Below is a screenshot of the `JPanel` when it first appears. Initially the scores shown are 0 and the `messageT` textField is empty.



Whenever the user presses the "Home/Away Team Score" buttons, the program:

- increments the `homeScore`/`awayScore` instance variable for the appropriate team based on the button pushed,

and

- sets the text in the `homeT`/`awayT` textfield to reflect the newly incremented score.

When the user presses the `gameOverB` button the program:

- displays in the `messageT` textfield one of the following messages:
  "Home Team Wins!" if the home team wins (i.e. has the higher score),
  "Away Team Wins!" if the away team wins (i.e. has the higher score),
  or
  "DRAW!" if both teams have the same score,

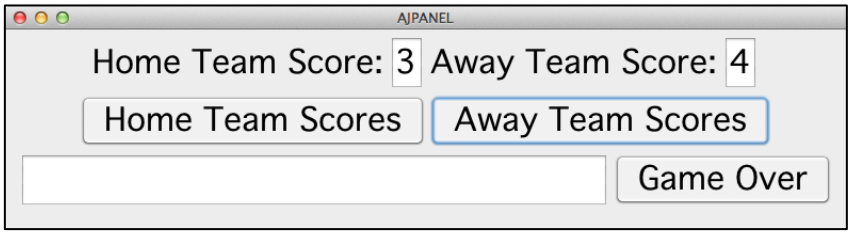
and

- resets both teams' scores to 0 and updates `homeT` and `awayT` textfields to reflect this.

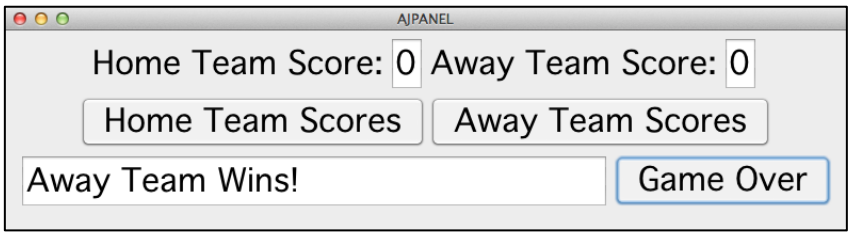
ID: .....

Below are two screenshots of the JPanel in use.

After some scores have been entered.



After the user has pressed the 'Game Over' button.



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.

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

                                                    
public class AJPanel extends JPanel                                {

      private JTextField homeT, awayT, messageT;
      private JButton homeB, awayB, gameOverB;
      private int homeScore, awayScore;

      public AJPanel() {
         homeT = new JTextField("0");
         awayT = new JTextField("0");
         messageT = new JTextField(20);
         homeB = new JButton("Home Team Scores");
         awayB = new JButton("Away Team Scores");
         gameOverB = new JButton("Game Over");

         add(new JLabel("Home Team Score:"));
         add(homeT);
         add(new JLabel("Away Team Score:"));
         add(awayT);
         add(homeB);
         add(awayB);
         add(messageT);
         add(gameOverB);
         homeScore = 0;
         awayScore = 0;
```

ID: .....

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

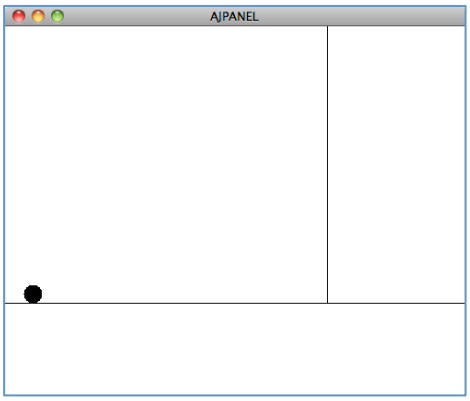
```
}
```

ID: .....

Question 11 (12 marks)

Complete the following `JPanel` which uses a `Timer` object. The `Timer` object is created with a delay of 50 milliseconds and the `Timer` starts as soon as the `JPanel` is created.

The `JPanel` models a simple projectile: a black ball, with size 20 pixels and with the initial top left position given by the constant `INITIAL_POSITION`, rests on the floor (with y coordinate given by `FLOOR_Y`). A wall (with x position `WALL_X`) is on the right of the JPanel. The screenshot below shows the initial state of the JPanel.

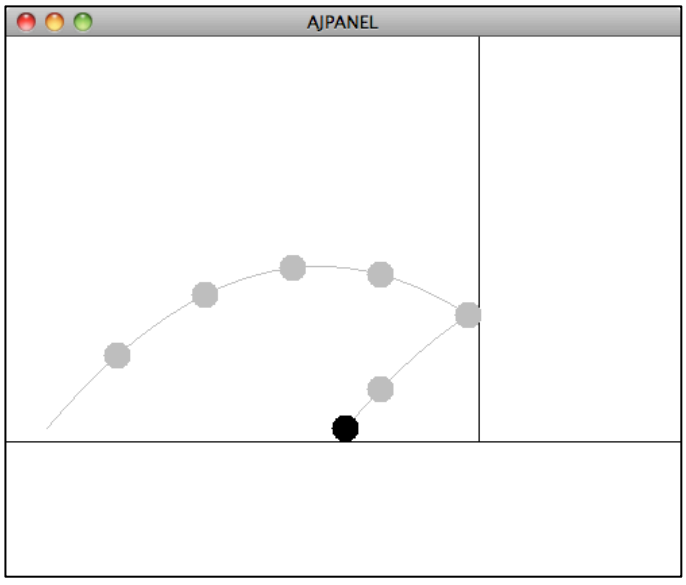


The number of pixels that the ball moves with every tick of the timer in the x and y directions respectively are given by `xVelocity` and `yVelocity`. Initially these values are 13 and 15. (Note: `yVelocity > 0` means the ball is moving *up* on the screen.)

Every time the timer ticks, the ball's `yVelocity` decreases by `GRAVITY`. Eventually the ball will fall and hit the floor. When the bottom of the ball touches the floor, the timer should stop. If the timer is stopped, hitting the UP arrow key should reset the ball to its initial position and velocities, and the timer should start again. If the UP key (or any other key) is pressed while the timer is running, nothing should happen.

If the right of the ball hits the wall, the ball should bounce back to the left, i.e., the `xVelocity` is reversed.

The screenshot below indicates the ball's flight on its way to its final position on the floor.



ID: .....

Note: You MUST use the variables and constants given in the code.

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

public class AJPanel extends JPanel                                    {

    public static final int FLOOR_Y = 300;
    public static final int WALL_X = 350;
    public static final int SIZE = 20;
    public static final Point INITIAL_POSITION = new
                                        Point(30, FLOOR_Y - SIZE);
    public static final int GRAVITY = 1;

    private int xVelocity, yVelocity;
    private Point position;
    private Timer t;

    public AJPanel() {

        reset();

    }
    private void reset() {
        position = new Point(INITIAL_POSITION);
        xVelocity = 13;
        yVelocity = 15;
    }
    public void paintComponent( Graphics g ) {
        super.paintComponent( g );
        g.drawLine(0, FLOOR_Y, 500, FLOOR_Y);
        g.drawLine(WALL_X, FLOOR_Y, WALL_X, 0);

    }

    public void actionPerformed(                          e) {
                                    //update the position of the ball

                                                  //update the yVelocity

    }
```

CONTINUED

ID: .....

```
                                //check if the ball has reached the wall









                               //check if the ball has reached the floor
```

```
    }

    public void keyPressed( KeyEvent e ) {
```

```


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

ID: .....

OVERFLOW PAGE

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

SURNAME: ..... FORENAMES: .....

OVERFLOW PAGE

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

SURNAME: ..... FORENAMES: .....

ROUGH WORKING (WILL NOT BE MARKED)

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

SURNAME: ..... FORENAMES: .....

ROUGH WORKING (WILL NOT BE MARKED)

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