THE UNIVERSITY OF AUCKLAND

SECOND SEMESTER, 2009
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 (/12)	**Q4** (/10)	**Q7** (/6)	**Q10** (/10)
Q2 (/8)	**Q5** (/10)	**Q8** (/7)	**Q11** (/10)
Q3 (/10)	**Q6** (/10)	**Q9** (/7)	**Total** /100

ID: .....

Question 1 (12 marks)

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

```
System.out.println(4 + 5 + " cats " + 3 + 5 * 2);
```

9 cats 310

(2 marks)

b) What is the output when the following code is executed.?

```
boolean a = true;
boolean b = false;
int x = 20;

System.out.println((x > (x / 5) * (x / 10)) && (a || !b));
```

true

(2 marks)

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

```
String name1 = new String("Winnie the Pooh");
String name2 = "Pooh";
String name3 = name1.substring(name1.length() - 4);

System.out.println("1: " + (name3 == name2));

System.out.println("2: " + (name3.equals(name2)));
```

```
1: false

2: true
```

(2 marks)

d) Complete the conditions in the following if-else-if-statement so the code produces the intended output:

```
System.out.print("Enter your grade (A, B, C or D): ");
String x = Keyboard.readInput();

if ( x.equals("A") ) {
    System.out.println("Wow – an A grade student!");
} else if ( x.equals("D") ) {
    System.out.println("Sorry, D is not a pass");
} else {
    System.out.println("You passed OK.");
}
```

(2 marks)

ID: .....

e) What does the following code output for an array, x, of six integers? Assume that the array has been declared and constructed, but that no other operations have been performed on the array before the code executes.

```
…

x[1] = 9;
x[3] = 5;
x[4] = 8;

System.out.println(x[x.length - 2] + x[1]);
```

17

(2 marks)

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

```
int[] myArray1 = {3, 5, 3, 6, 8};
int[] myArray2;

myArray2 = new int[myArray1.length];

for (int i = myArray1.length - 1; i >= 0; i--) {
    myArray2[i] = myArray1[i];
}

myArray2[4] = 9;

System.out.println("1: " + myArray1[4]);
System.out.println("2: " + myArray2[3]);
```

1: 8

2: 6

(2 marks)

ID: .....

Question 2 (8 marks)

Question 2 is based on the following class.

The MyClass class definition below is incomplete (incomplete sections are denoted by "..."):

```
public class MyClass {

    private int x;

    // Constructor method:
    public  … (int x) {
        …
    }

    // Mutator method for x
    public … setX( … ) {
        …
    }

    // Accessor method for x
    public int getX() {
        …
    }
}
```

a) Complete the constructor method for the class MyClass.

```
public MyClass (int x) {

    this.x = x;

}
```

(2 marks)

b) Define the accessor method, getX(), for the class MyClass.

```
public int getX () {

    return x;

}
```

(2 marks)

ID: .....

c) Complete the mutator method, setX(), for the class MyClass.

```
public void setX ( int x ) {

    this.x = x;

}
```

(2 marks)

d) What is the output when the following code is executed?

```
MyClass a = new MyClass(2);
MyClass b = a;

a.setX(3);
b.setX(a.getX() + 1);

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

```
4 4
```

(2 marks)

ID: .....

Question 3 (10 marks)

a) What is the output when the following start() method is executed?

```
public void start() {

     int result = calculate(calculate(15, 3), 2);

     System.out.println(result);
}

private int calculate(int value1, int value2) {
     return value1 / value2 ;
}
```

2

(2 marks)

b) What is the output when the following start() method is executed?

```
public void start() {
     String output = getStringOutput(2, 3.6, "6");

     System.out.println(output);
}

private String getStringOutput(int num1, double num2,
                                          String num3) {

     return num1 + "" + Double.parseDouble(num2 + num3);
}
```

23.66

(2 marks)

c) What is the output when the following start() method is executed?

```
public void start() {
     int a = calculate(7, 4, 8);

     System.out.println(a);
}

private int calculate(int num1, int num2, int num3) {
     int a = Math.min(num1, num2);
     int b = Math.min(num2, num3);

     return Math.min(a, b);
}
```

4

(2 marks)

d) Complete the output when the `start()` method below is executed.

```
public void start() {
     int a = 16;
     int b = doSomething(a, 5);
     System.out.println("a: " + a + " b: " + b);
}

private int doSomething(int a, int num) {
     int result = (a + num) / 2;
     if (result > num) {
          return a - 5;
     }
     return result;
}
```

a: 16 b: 11

(2 marks)

ID: .....

e) 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    private  class Program {

2         public void start() {
3              int a = 4;
4              method(a);
5         }

6         private void method(int x) {

7              int b = x;
8              a = b + 1;
9              System.out.println(a + " " + b);
10        }
11   }
```

Line number of the code
which causes the compile error: **line 8**

(2 marks)

ID: .....

Question 4 (10 marks)

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

```
Point p1 = new Point(4, 3);
Point p2 = new Point(3, 5);

p1.move(p2.y, p1.x);
p2.translate(8, 3);

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

```
1: 5, 4

2: 11, 8
```

(2 marks)

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

```
Point p1 = new Point(5, 2);
Point p2 = new Point(5, 1);
Point p3 = p2;

p2 = new Point(p3.x, p3.y + 1);

System.out.println("1: " + p1.equals(p2));
System.out.println("2: " + (p1 == p2));
```

```
1: true

2: false
```

(2 marks)

ID: .....

c) Complete the code below so that the output is: `Yes`

Any rectangle which contains the Point 20, 30 is correct, e.g.,

```
Rectangle rect;

rect = new Rectangle( 10, 20, 50, 50 );

Point p = new Point(20, 30);

if (rect.contains(p)) {
     System.out.println("Yes");
} else {
     System.out.println("No");
}
```

(2 marks)

d) Complete the code below which calculates the x and y values of the centre position of the Rectangle, rect.

```
Rectangle rect = new Rectangle(…, …, …, …);
```

```
int centreX = rect.x + rect.width / 2;


int centreY = rect.y + rect.height / 2;
```

(2 marks)

ID: .....

e) What is the output when the `start()` method below is executed?

```
public void start() {
     Point p1 = new Point(3, 5);
     doSomething(p1);
     System.out.println("1: " + p1.x + ", " + p1.y);

}

private void doSomething(Point p) {
     p.y = p.y + 3;
     System.out.println("2: " + p.x + ", " + p.y);
}
```

```
2: 3, 8
1: 3, 8
```

(2 marks)

ID: .....

Question 5 (10 marks)

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

```
int x = 9;
int y = 3;
while (x > y) {
    System.out.print("*");

    if (x > 7) {
        y--;
    }
    x--;
    y++;
}
```

****

(5 marks)

b) Consider the following method that copies an array of objects:

```
private Thing[] copyMyClassArray(Thing[] a) {

    Thing[] b = new Thing[a.length];

    for (int i = 0; i < a.length; i++) {
        b[i] = a[i];
    }
    return b;
}
```

Not counting the two arrays `a` and `b`, state how many new objects are created by the method. In your own words, justify why this is so.

None. The code 'b[i] = a[i];' assigns the address of the String object, a[i], to the array element, b[i]. The code does not create a new object.

(5 marks)

ID: .....

Question 6 (10 marks)

a) Complete the printDaysAndHours() method which is passed an int parameter representing the number of hours. The method calculates and prints the equivalent number of days (one day is 24 hours) and hours. The following call to the completed printDaysAndHours() method

```
printDaysAndHours(260);
```

produces the following output:

```
260 hours is 10 days and 20 hours.
```

```
private void printDaysAndHours(int hours) {

  int days;

  System.out.print(hours + " hours is ");


   days = hours / 24;
   hours = hours % 24;


 System.out.println(days + " days and " + hours +
                                          " hours.");

}
```

(4 marks)

b) Complete the following luckyDip() method which is passed two int parameters, the lower bound and the upper bound. The method generates two random integers between the lower and the upper bounds (inclusive), sorts them in ascending order and returns a String containing the two random numbers separated by ", ". The following two lines of code (using the completed luckyDip() method)

```
String result = luckyDip(10, 20);
System.out.println(result);
```

would print a String containing two numbers between 10 and 20 in ascending order. For example, the output might be: "12, 19" or "15, 16" or "10, 20" or "14, 14", etc.

```
private String luckyDip(int lowerBound,
                                        int upperBound) {

    String sortedNumbers =   "";



    int range = upperBound – lowerBound + 1;
    int num1 = (int) (Math.random() * range)
                                        + lowerBound;

    int num2 = (int) (Math.random() * range)
                                        + lowerBound;



    int min = Math.min(num1, num2);
    int max = Math.max(num1, num2);

    sortedNumbers = "" + min + ", " + max;



    return sortedNumbers;
}
```

(6 marks)

ID: .....

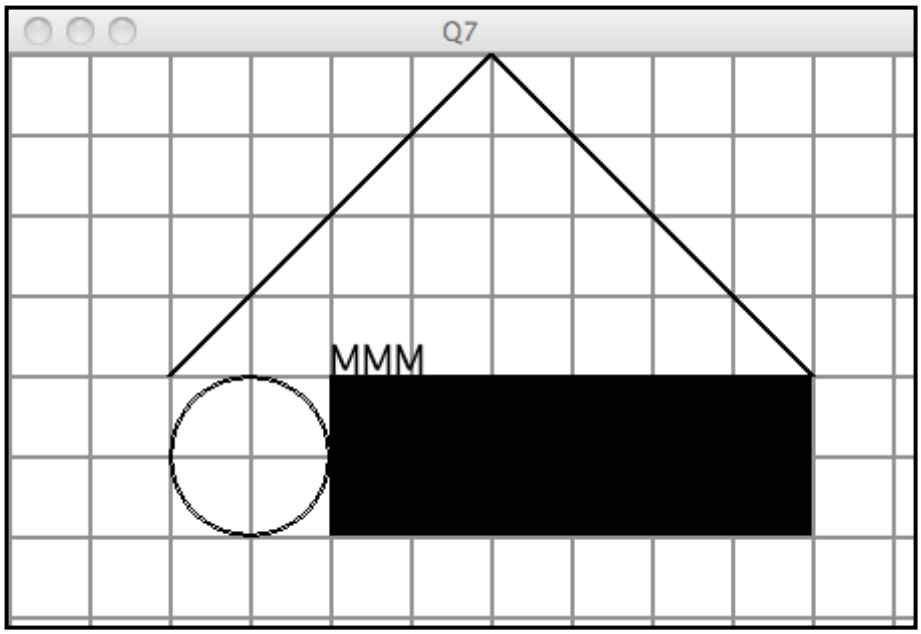
Question 7 (6 marks)

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);

        drawThing(g, 20, 40, 20);
   }

   public void drawThing(Graphics g, int x, int y,
                                                int size) {

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

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

        g.drawLine(x + size * 2, y - size * 2, x + size * 4, y);

        g.fillRect(x + size, y, size * 3, size);

        g.drawString("MMM", x + size, y);

     }
}
```

ID: .....



(6 marks)

ID: .....

Question 8 (7 marks):

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

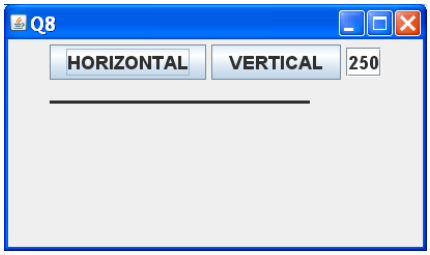
lengthT – a JTextField which contains an int value,
horizontalB – a JButton displaying the String "HORIZONTAL",
verticalB – a JButton displaying the String "VERTICAL".

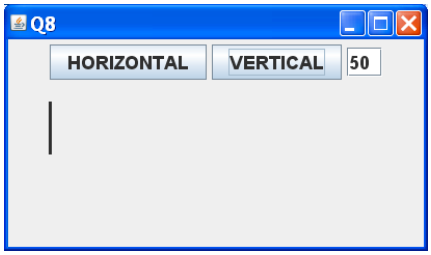
To the right is a screenshot of the JPanel when it is first displayed. The lengthT JTextField displays the number 100 and a vertical line 100 pixels in length is displayed starting at position (40, 100).

When the user presses the "HORIZONTAL" JButton, a horizontal line is drawn in the JPanel from position (40, 100). The length of the horizontal line is given by the value in the JTextField, lengthT. When the user presses the "VERTICAL" JButton, a vertical line is drawn in the JPanel from position (40, 100). The length of the vertical line is given by the value in the JTextField, lengthT. The user may enter any length in the JTextField lengthT before pressing one of the two JButtons.

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.

The screenshots below show two examples of the JPanel when the user enters a value in the lengthT JTextField and presses either the "HORIZONTAL" or the "VERTICAL" JButton.






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

public class AJPanel extends JPanel implements ActionListener {

  private JTextField lengthT;
  private JButton horizontalB, verticalB;
  private boolean isHorizontal;

  public AJPanel() {
      lengthT = new JTextField("100");

      horizontalB = new JButton("HORIZONTAL");
      verticalB = new JButton("VERTICAL");
```

CONTINUED

ID: .....

```
        add(horizontalB);
        add(verticalB);
        add(lengthT);
```

```
        horizontalB.addActionListener(this);
        verticalB.addActionListener(this);
```

```
}
```

```
 public void actionPerformed ( ActionEvent e) {

        if (e.getSource() == horizontalB) {
            isHorizontal = true;
        } else {
            isHorizontal = false;
        }

        repaint();
}
```

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

```
        final int START_X = 40;
        final int START_Y = 100;

        int length;

        length =
              Integer.parseInt(lengthT.getText());

        if (isHorizontal) {
            g.drawLine(START_X, START_Y,
                       START_X + length, START_Y);
        } else {
            g.drawLine(START_X, START_Y, START_X,
                               START_Y + length);
        }
```

```
}
```

```
}
```

(7 marks)

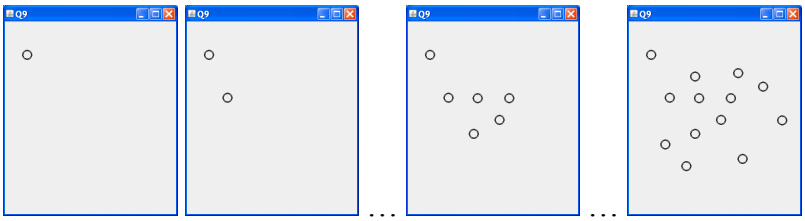
ID: .....

Question 9 (7 marks)

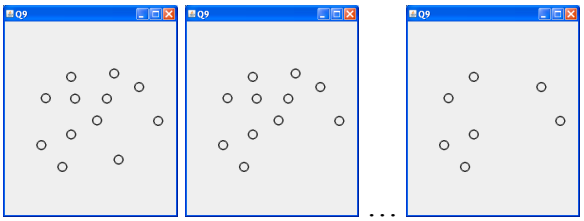
The following JPanel responds to MouseEvents.

Initially, the JPanel is empty. Whenever the user presses the mouse in the JPanel in a location which does not already display a circle, a new circle of size 20 pixels is drawn with the mouse press position as the centre. If the user presses the mouse inside an existing circle the circle should disappear. The JPanel can display up to 1000 circles.

The following screen shots show the behaviour of the completed JPanel. First the user presses the mouse in various positions in the JPanel:



then the user presses the mouse inside some of the existing circles:



The following 14 statements (or sets of statements) are part of the code. Place each **number** in the correct place in the `AJPanel` definition on the next page so that the `JPanel` executes as described above.

```
1.  MouseEvent e
2.  implements MouseListener
3.  repaint();
4.  squares = new Rectangle[1000];
    isVisible = new boolean[1000];
5.  addMouseListener(this);
6.  int x = e.getX();
    int y = e.getY();
7.  public void mouseClicked(MouseEvent e) {}
8.  isVisible[upTo] = true;
    upTo++;
9.  extends JPanel
10. import java.awt.event.*;
11. for(int i = 0; i < upTo; i++) {
      if (isVisible[i]) {
        g.drawOval(squares[i].x, squares[i].y,
                        squares[i].width,squares[i].height);
      }
    }
12. boolean isInVisibleCircle =
                          mousePressIsInVisibleCircle(x, y);
13. private boolean[] isVisible;
14. upTo = 0;
```

ID: .....

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;  //10
                                                    //9      2
public class Q9JPanel extends JPanel implements MouseListener {
        private Rectangle[] squares;
        private boolean[] isVisible; //13
        private int upTo;

        public Q9JPanel() {
                addMouseListener(this);                 //5
                squares = new Rectangle[1000];          //4
                isVisible = new boolean[1000];
                upTo = 0;                               //14
        }

        public void mousePressed(MouseEvent e) { //1
                int x = e.getX();                       //6
                int y = e.getY();
                                                        //12
                boolean isInVisibleCircle = mousePressIsInVisibleCircle(x, y);

                if (! isInVisibleCircle && upTo < 1000) {
                        squares[upTo] = new Rectangle(x - 10, y - 10, 20, 20);
                        isVisible[upTo] = true;         //8
                        upTo++;
                }

                repaint();                              //3
        }
        private boolean mousePressIsInVisibleCircle(int x, int y) {
                Point p = new Point(x, y);

                for(int i = 0; i < upTo; i++) {
                        if (isVisible[i] && squares[i].contains(p)) {
                                isVisible[i] = false;
                                return true;
                        }
                }
                return false;
        }
        public void paintComponent(Graphics g) {
                super.paintComponent(g);
                for(int i = 0; i < upTo; i++) {         //11
                        if (isVisible[i]) {
                                g.drawOval(squares[i].x, squares[i].y,
                                                squares[i].width, squares[i].height);
                        }
                }
        }
```

CONTINUED

ID: .....

```
        public void mouseClicked(MouseEvent e) {}  //7
        public void mouseEntered(MouseEvent e) {}
        public void mouseExited(MouseEvent e) {}
        public void mouseReleased(MouseEvent e) {}

}
```

(7 marks)

ID: .....

Question 10 (10 marks)

Consider the following code:

```
Car c = new Car();

c.setSpeed(80); // Set the car's speed in km/h

c.drive(30);    // Drive for 30 minutes
int distance = c. getDistance();
         // Output distance driven – 40 km
System.out.println("Distance: " + distance + "km");
System.out.println("Speed is:" + c.getSpeed() + "km/h");


c.drive(15);   // Drive for another 15 minutes
         // Output distance driven – 60 km
System.out.println("Distance: " + c.getDistance() + "km");
```

Your task is to complete the definition of a Car class which can be used with the above code, based on the partially implemented class below.

To complete the class:

- Complete the empty instance methods given
- Identify which instance methods are used above, and implement them in the class
- Implement and initialize an additional instance variable.

```
public class Car {

    private int speedKmPerH;

    private int distanceDriven;


    public Car() {
        speedKmPerH = 0;
        distanceDriven = 0;
    }

    public void setSpeed(int s) {
        speedKmPerH = s;
    }

    public int getSpeed() {
        return speedKmPerH;

    }
```

ID: .....

```
  public int getDistance() {
    distanceDriven;
  }


  public void drive(int minutes) {
    distanceDriven = distanceDriven +
                     speedKmPerH * minutes / 60;
  }
```

}

(10 marks)

ID: .....

Question 11 (10 marks)

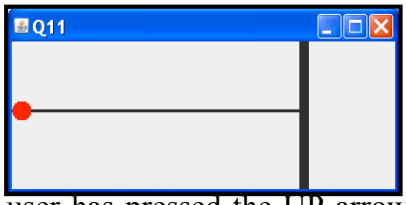
You are required to complete the following JPanel which uses a Timer object. The Timer object is created with a delay of 100 milliseconds. Initially, a filled red circle is displayed on the left hand side of the JPanel, a vertical line is displayed down the right hand side of the JPanel and a horizontal line across the JPanel. The Timer starts when the user presses the UP arrow key. The red circle then begins to move along the horizontal line towards the vertical line.

Each time the Timer ticks, the red circle moves **ten** pixels. When the right hand side of the circle reaches the vertical line on the right hand side of the JPanel then the red circle starts moving to the left along the horizontal line. When the left hand side of the circle reaches the left hand edge of the JPanel the circle starts moving to the right again and so on.

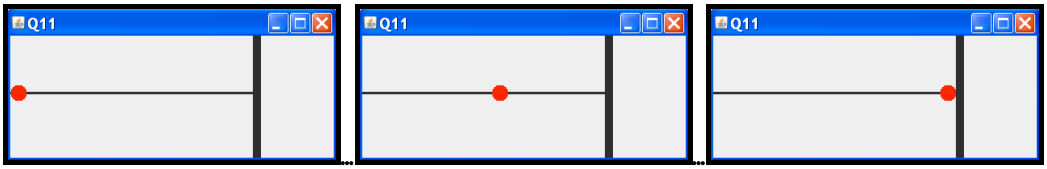
By pressing the UP and DOWN arrow keys, the user controls the Timer. When the user presses the UP arrow key, the Timer should start and the circle should start moving in its current direction. When the user presses the DOWN arrow key, the Timer should stop. When the user presses the UP arrow key again, the Timer starts again and the circle continues moving to the right/left, etc.

Below are some screenshots of the JPanel in action.

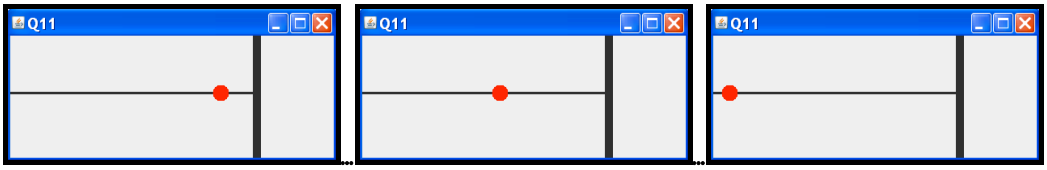
The screenshot on the right shows the JPanel when it is first displayed.



The next three screenshots show the JPanel after the user has pressed the UP arrow key and the circle has started moving to the right.



The next three screenshots show the circle moving to the left after it has reached the vertical line.



Notes:

The x value of the line on the right hand side of the JPanel is given by the constant:

```
public static final int RIGHT_HAND_SIDE = 300;
```

The starting position of the red circle is given by the constant:

```
public static final Rectangle BLOB_INITIAL_POSITION = new
                                        Rectangle(0, 60, 20, 20);
```

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

ID: .....

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

public class AJPanel extends JPanel implements ActionListener, KeyListener {

     //The starting position of the red circle
public static final Rectangle BLOB_INITIAL_POSITION = new
                                        Rectangle(0, 60, 20, 20);
public static final int RIGHT_HAND_SIDE = 300;
public static final int LINE_Y = 70;

private Rectangle blob;
private boolean isMovingRight;
private Timer t;

public AJPanel() {
     isMovingRight= true;
     blob = new Rectangle(BLOB_INITIAL_POSITION);

     t = new Timer(100, this);
     addKeyListener(this);
}

public void actionPerformed(ActionEvent e) {

  if (isMovingRight) {
    blob.x = blob.x + 10;

    if(blob.x + blob.width >= RIGHT_HAND_SIDE){
        isMovingRight = false;
        blob.x = RIGHT_HAND_SIDE – blob.width;
    }
  } else {
        blob.x = blob.x - 10;

        if (blob.x < 0) {
            blob.x = 0;
            isMovingRight = true;
        }
  }
  repaint();
}
```

ID: .....

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

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

```
}

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

            //the thin rectangle displayed on the
            //right hand side of the window

    g.fillRect(RIGHT_HAND_SIDE, 0, 10, 150);

            //the horizontal line
    g.drawLine(0, LINE_Y, RIGHT_HAND_SIDE, LINE_Y);
```

```
    g.setColor(Color.RED);

    g.fillOval(blob.x, blob.y, blob.width,
                                    blob.height);
```

```
}

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)