CompSci 101 Exam
Second Semester 2005

ANSWER BOOKLET

Surname ...ANSWERS.....

Forenames ...MODEL.....

Student ID .....

Login name(UPI) .....

Examiner to complete:

Question	Mark
1 – 22	**55** **(/55)**
23	**6** **(/6)**
24	**16** **(/16)**
25	**7** **(/7)**
26	**16** **(/16)**
Total	**100** **(/100)**

ID.....

SECTION A: MULTIPLE CHOICE QUESTIONS

Each question in this section is worth 2.5 marks. **Circle** the letter corresponding to your choice. There is only one correct answer for each question. If you make a mistake, mark a cross through your wrong choice and circle your next alternative.

1.	a	b	c	d	**e**
2.	a	**b**	c	d	e
3.	a	**b**	c	d	e
4.	a	b	**c**	d	e
5.	a	**b**	c	d	e
6.	a	**b**	c	d	e
7.	a	b	c	**d**	e
8.	a	b	c	d	**e**
9.	**a**	b	c	d	e
10.	**a**	b	c	d	e
11.	a	b	**c**	d	e

ID.....

12.	a	b	c	**d**	e
13.	a	**b**	c	d	e
14.	a	b	**c**	d	e
15.	**a**	b	c	d	e
16.	a	b	c	d	**e**
17.	a	b	c	**d**	e
18.	a	b	c	**d**	e
19.	a	**b**	c	d	e
20.	a	b	**c**	d	e
21.	**a**	b	c	d	e
22.	**a**	b	c	d	e

SECTION B

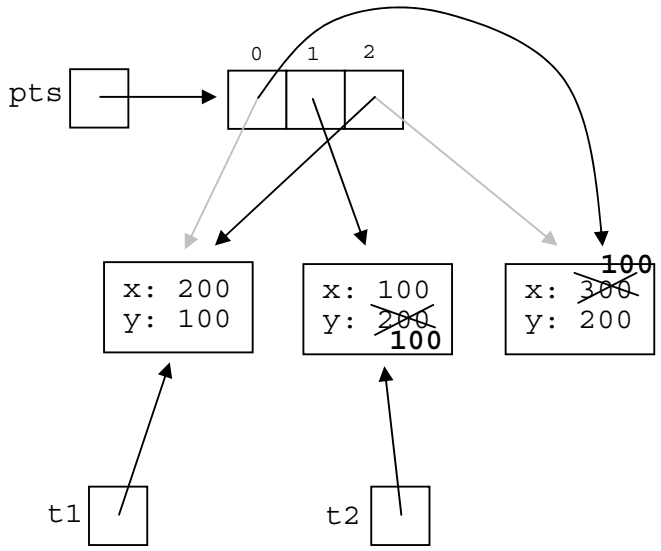
Write your answers in the spaces provided.

23. What is the output produced by the following code?

```
100 , 200
100 , 100
200 , 100
```

(6 marks)

You may use the space below to draw a diagram of the array of `Points`. If your answer above is incorrect, then you may receive some partial credit for a clearly drawn diagram.



24. Complete the `RacingCarPanel` and `RacingCar` classes below.

```
import java.awt.*;

public class RacingCar {

     private int size;

     /**
      * This is the constructor of the class. View how it is being
      * used in the RacingCarPanel class and fill in the
      * appropriate code.
      */
     public RacingCar( [          int initialSize          ] ) {

          [
           size = initialSize;
          ]
     }

     /**
      * This is a mutator method used by the RacingCarPanel class
      * to update the size variable.
      */
     public [     void setSize     ] ( [     int newSize     ] ){

          [
           size = newSize;
          ]
     }
```

ID.....

```
/*
 * This method draws the car on the screen according to
 * the size the user has chosen.
 */
public void draw(Graphics g) {

    // Draw the main body of the car
    g.setColor(Color.red);
    g.fillRect(size, size*3, size*6, size*2);
    g.fillRect(size*4, size*2, size*2, size);

    // Draw the wheels
    g.setColor(Color.black);
    g.fillOval(size*2, size*5,size, size);
    g.fillOval(size*5, size*5,size, size);

}
}
```

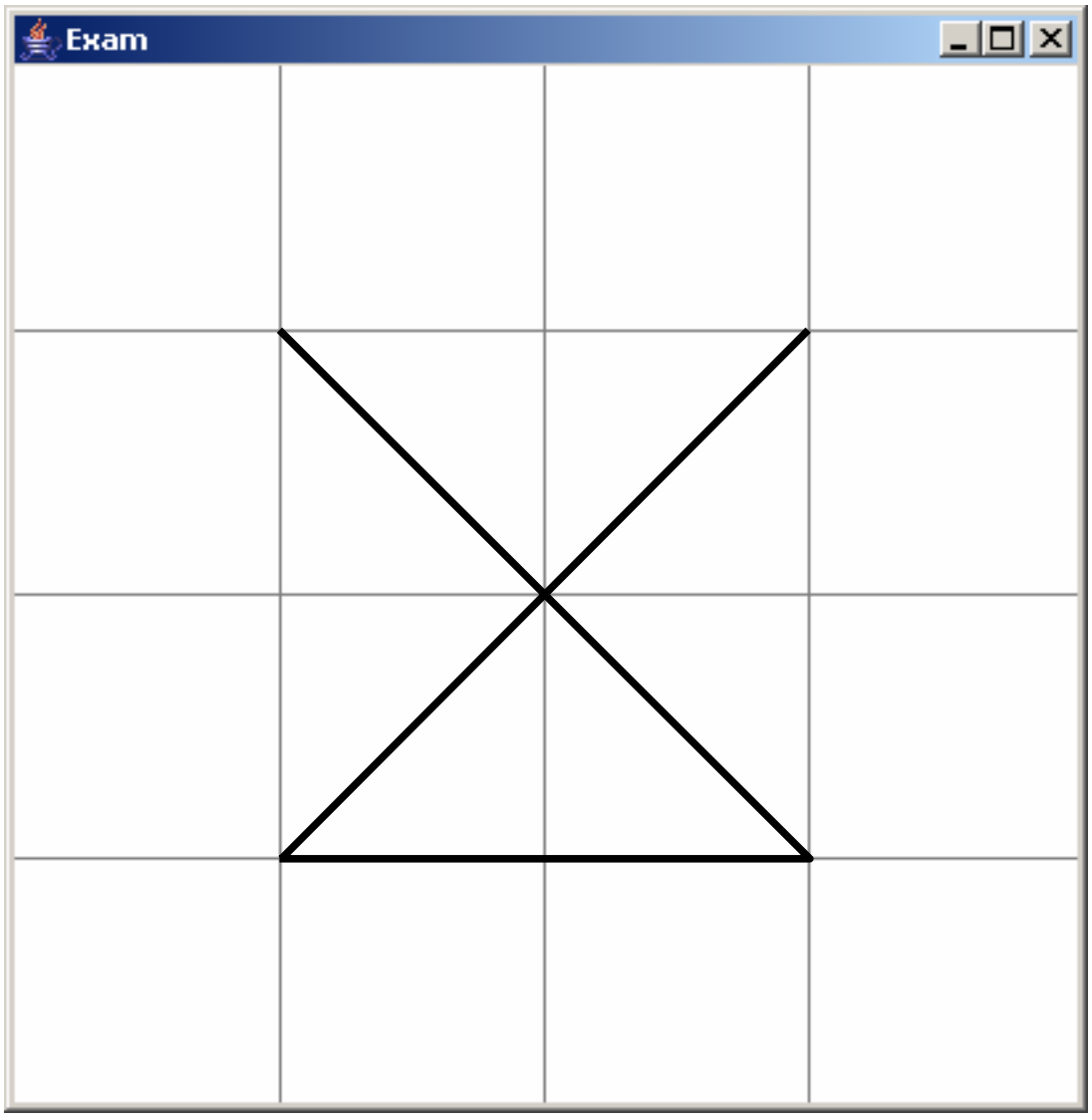
ID.....

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

public class RacingCarPanel  extends JPanel implements
                                          ActionListener {

     private JTextField sizeField;
     private JButton drawButton;
     private RacingCar car;

     public RacingCarPanel(){
          sizeField = new JTextField("30");
          drawButton = new JButton("draw");

          drawButton.addActionListener(this);

          add(sizeField);
          add(drawButton);
          car = new RacingCar(30);
     }

     public void paintComponent(Graphics g){
          super.paintComponent(g);
          car.draw(g);
     }

     /**
      * This method is added to process events that are
      * generated by the button.
      */
     public void  actionPerformed  (  ActionEvent  e) {

          int newSize = Integer.parseInt( sizeField.getText() );
          car.setSize( newSize );
          repaint();

          }
     }
}
```

(16 marks)

25. Draw the output produced by this program in the window provided below. You can assume the window is 400 pixels wide and 400 pixels high.

NOTE: Grid lines are displayed for you on the window to help you draw your diagram accurately. Each grid square below is 100 pixels high and 100 pixels wide.



(7 marks)

26. Complete the `Circles` class in the space below.

```
import java.awt.*;

public class Circles {

     private Point[] pos;
     private int numCircles;

     public Circles() {
          pos = new Point[1000];
          numCircles = 0;
     }

     public void add(int x, int y) {
          pos[numCircles] = new Point(x, y);
          numCircles++;
     }

     public void deleteLast() {
          if (numCircles > 0) {
               numCircles--;
          }
     }

     public void draw(Graphics g) {
          for (int i = 0; i < numCircles; i++) {
               g.drawOval(pos[i].x-10, pos[i].y-10, 20, 20);
          }
     }
}
```

(16 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.....

ROUGH WORKING
(Will not be marked)

ID.....

ROUGH WORKING
(Will not be marked)