

Week 12

Applets

EXPLORING
Java



Week 12

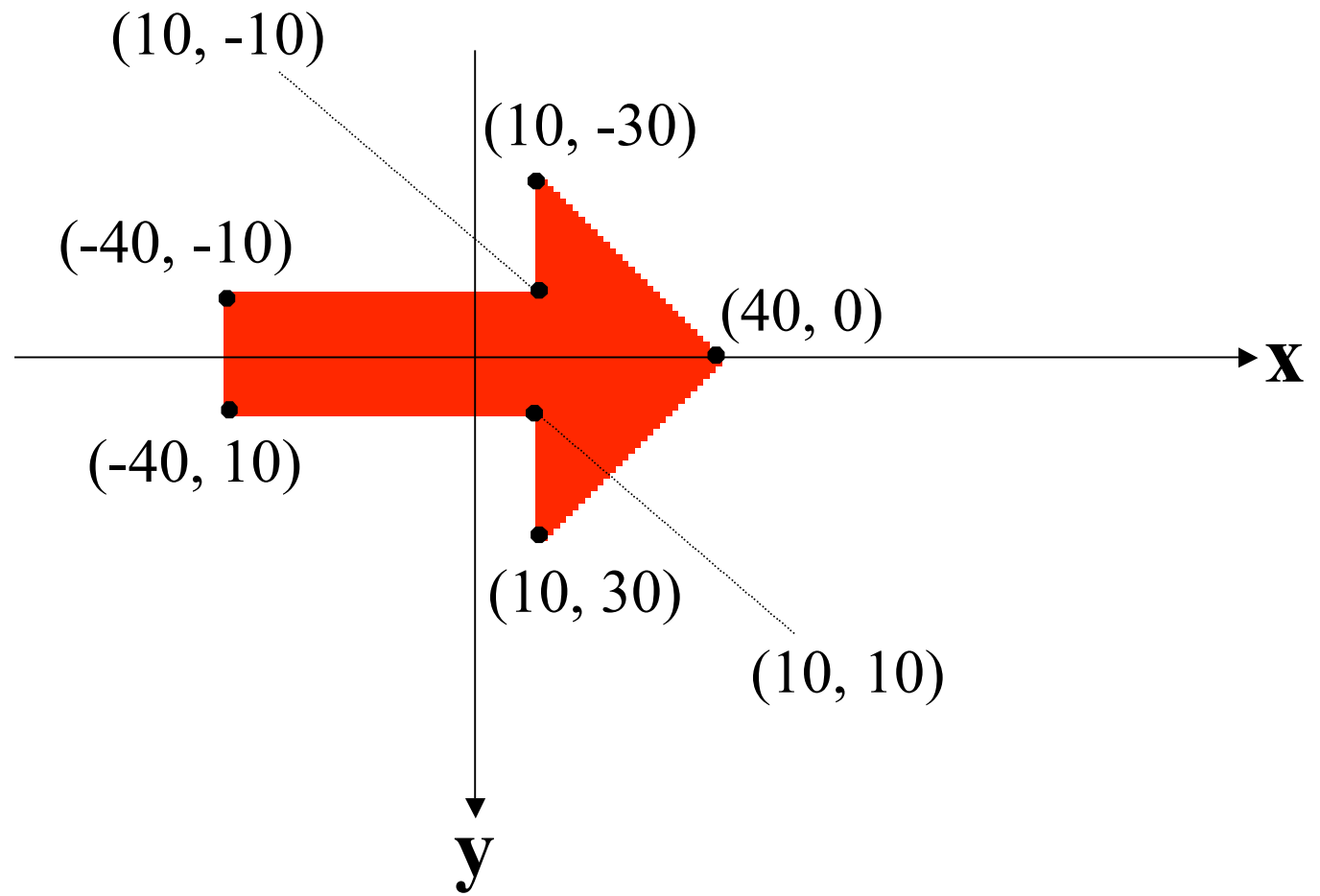
Lecture: Applets &

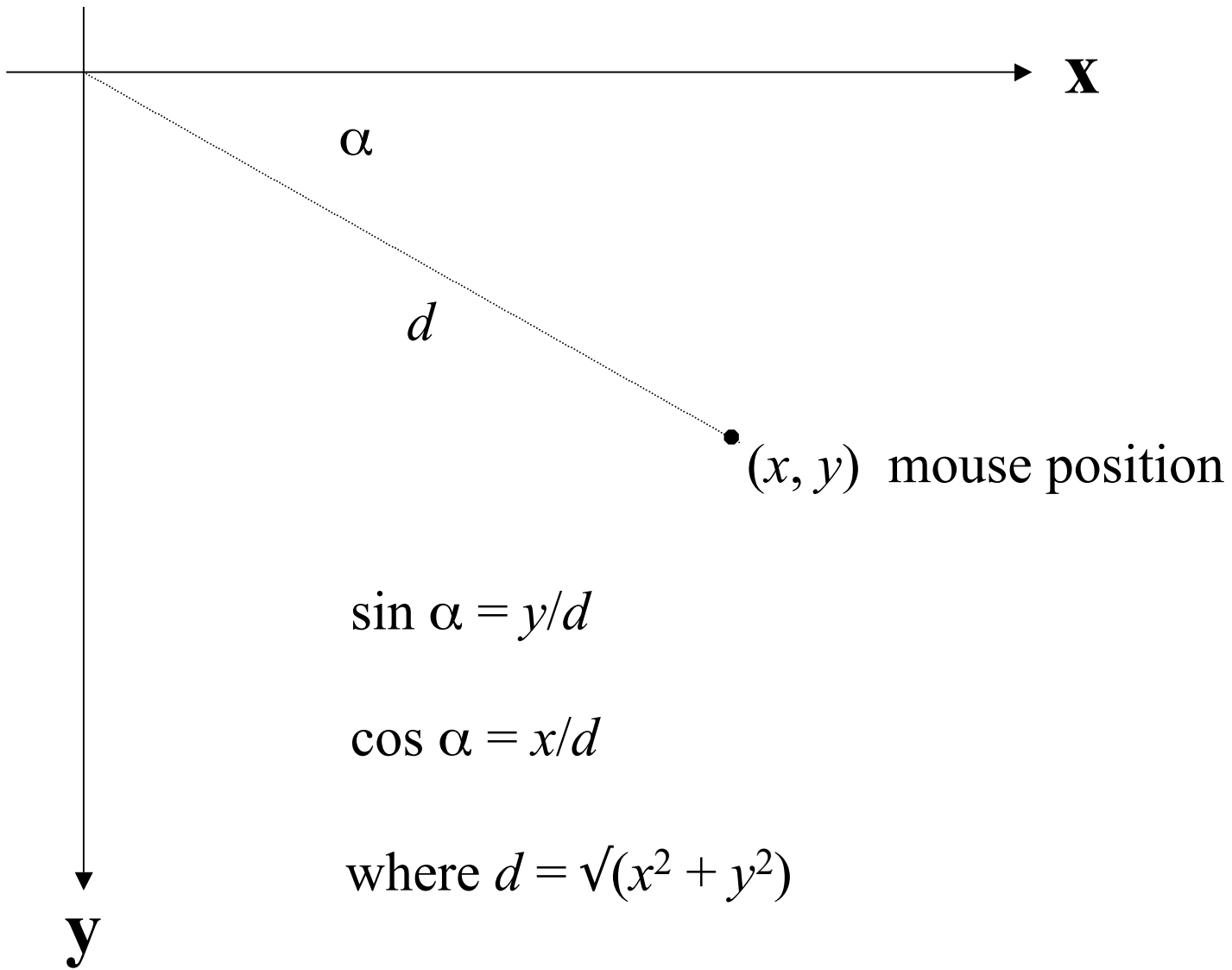
Working the sample exam

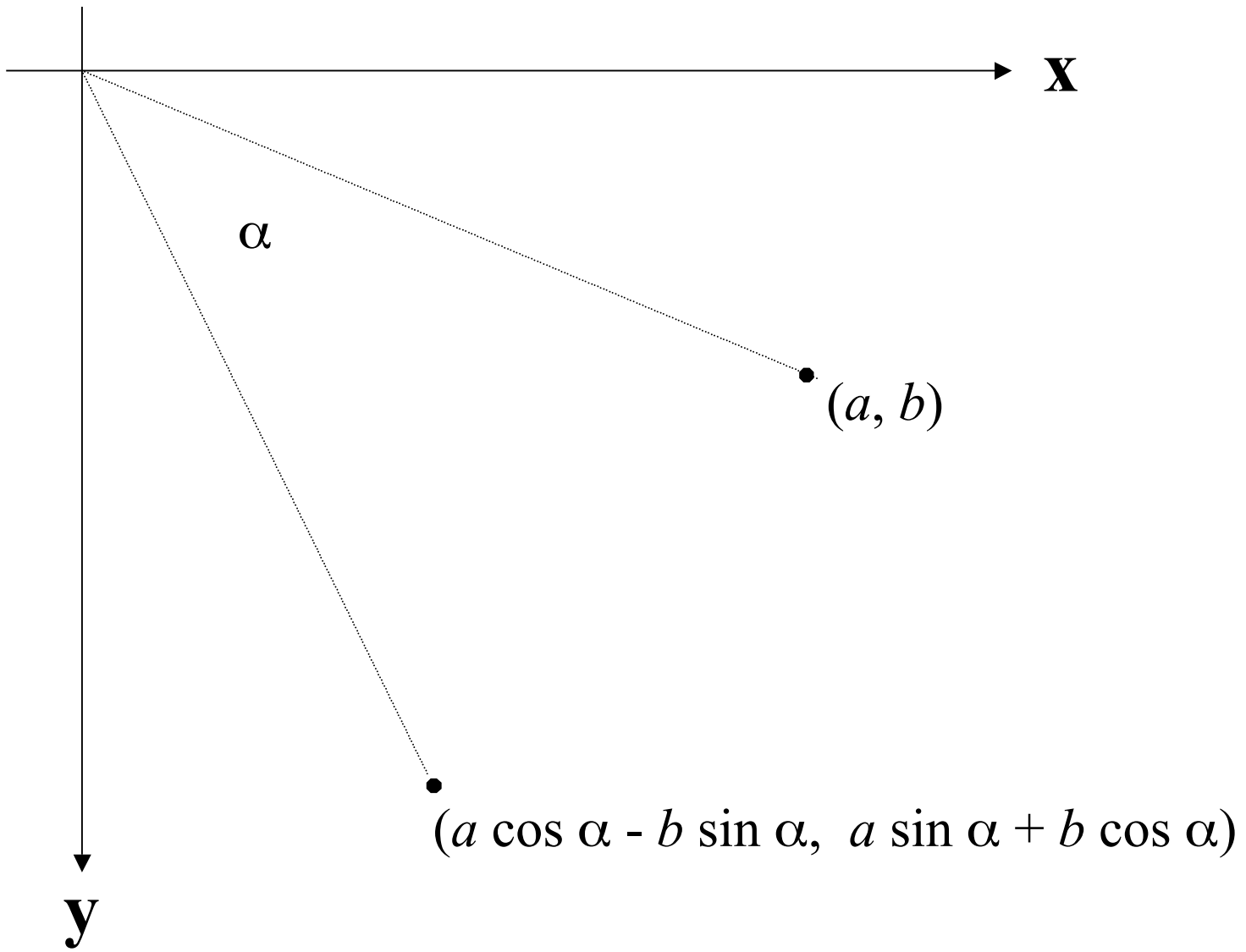
Java Genesis:

–Ch12: Applets

Lab Assessments 9 and 10







```

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

public class PointerApplet extends JApplet {

    private PointerPanel panel = new PointerPanel( );

    public void init ( ) {
        Container c = getContentPane( );
        JButton tog = new JButton("toggle style");
        tog.addActionListener(new ActionListener( ) {
            public void actionPerformed(ActionEvent e) {
                panel.toggleStyle( );
            }
        });
        c.add(panel);
        JPanel pBot = new JPanel( );
        pBot.add(tog);
        c.add(pBot, "South");
    }
}

```

```

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

public class PointerPanel extends JPanel {

    private double sin = 0, cos = 1;
    private boolean isSolid = true;

    public PointerPanel ( ) {
        setBackground(Color.white);
        addMouseListener(new MouseMotionAdapter ( ) {
            public void mouseMoved(MouseEvent e) {
                int x = e.getX( ) - 250;
                int y = e.getY( ) - 250;
                double dist = Math.sqrt(x*x + y*y);
                sin = y/dist;
                cos = x/dist;
                repaint( );
            }
        });
    }
}

```

```

public void paintComponent (Graphics g) {
    super.paintComponent(g);
    int [ ] xs = {-40, 10, 10, 40, 10, 10, -40};
    int [ ] ys = {-10, -10, -30, 0, 30, 10, 10};
    int [ ] rotXs = new int [7];
    int [ ] rotYs = new int [7];
    for (int i=0; i<7; i++) {
        rotXs[i] = (int)(xs[i]*cos - ys[i]*sin);
        rotYs[i] = (int)(xs[i]*sin + ys[i]*cos);
    }
    g.setColor(Color.red);
    g.translate(250, 250);
    if (isSolid) g.fillPolygon(rotXs, rotYs, 7);
    else g.drawPolygon(rotXs, rotYs, 7);
}

public void toggleStyle ( ) {
    isSolid = !isSolid;
    repaint( );
}
}

```

```
<HTML>
<HEAD>
<!-- Generated by Kawa IDE -->
<TITLE>Applet title</TITLE>
</HEAD>
<BODY>
<H1>Pointing at the mouse</H1>
<HR>
<APPLET CODE=PointerApplet.class WIDTH=500 HEIGHT=500></APPLET>
<HR>
</BODY>
</HTML>
```

How to pass this course

You will pass this course if

- you get 50 or more for the combined assessment (labs, assignments, prac exam and m-c exam)
- and
 - you get 15 or more for the prac exam
 - or you get 25 or more for the prac and m-c exams combined
- **See the course profile for full details.**

How Grades are Determined

Let E be the mark for exams (out of 50).

Let T be the total mark (out of 100).

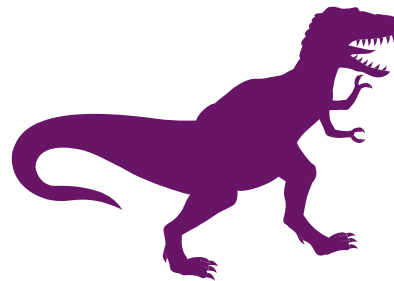
- For a 7 require $E \geq 45$ and $T \geq 90$.
- For a 6 require $E \geq 40$ and $T \geq 80$.
- For a 5 require exams passed and $T \geq 70$.
- For a 4 require exams passed and $T \geq 50$.

The Prac Exam

- The exam starts with half an hour of preparation (without the computer) followed by 2 more hours of working with the computer.
- As each question is completed, submit the required code for that question using the on-line submission system (record the Submission ID on the exam paper).

more Prac Exam

- **For each question make sure you submit the required *.java* files, namely, those that you have modified as specified by the question.**



yet more Prac Exam

- The exam is divided into 3 questions.
- Applets (Chap 12), multi-dimensional arrays (Sec 5.4) and interfaces (Sec 10.5) will not be examined.
- You may bring any written material into the exam room, but no floppy disks.
- You will have a new exam account; consequently H-drive will not be available.

even yet more Prac Exam

- The Prac Exam is an opportunity to show what you have learned in the course.
- At all times the marker will be on your side, trying to award you marks if at all possible.

Multiple-Choice Exam

- Make sure you carefully record your selected options on the answer sheet supplied: these sheets will be marked by machine.
- Accurately record your name and student number on the answer sheet.
- Also indicate your selected option for each question on the question paper itself. This paper will be collected after the exam.

more Multiple-Choice Exam

- The exam start with 10 minutes for perusal followed by 1 hour for working.
- The paper will consist of 20 questions, each with 5 option, precisely **one** of which is correct:
do not mark more than one option.
- Marks are not deducted for incorrect answers, so if you're unsure of an answer, guess!

```
import java.io.*;
import genesis.*;

public class AverageAndCopy {

    public static void main (String [ ] args)
        throws IOException {
        int [ ] data = {5, 12, -34, 78, -92, 10};
        double average =
            averageOfPositives(data);
        Transcript.println(average);
        copyToFile("data.txt", data, 3);
    }
}
```

```
public static double averageOfPositives
                    (int [ ] intArray) {
    // ... Question 1(a)
    // this method returns the average of
    // the positive integers in the array
    // ...
}
```

```
public static void copyToFile
                    (String name, int [ ] xs, int w)
                    throws IOException {
    // ... Question 1(b)
    // this method outputs the integers in
    // the array to a named file, with the
    // integers output w to a line with a
    // space after each integer on a line
    // ...
}
```

```
}
```

```
public static double averageOfPositives
    (int [ ] intArray) {
    double sum = 0;
    int count = 0;
    for (int i=0; i<intArray.length; i++) {
        if (intArray[i] > 0) {
            sum = sum + intArray[i];
            count++;
        }
    }
    if (count == 0) return 0;
    else return sum/count;
}
```

```

public static void copyToFile
    (String name, int [ ] xs, int w)
        throws IOException {
    FileWriter fw = new FileWriter(name);
    PrintWriter pw = new PrintWriter(fw);
    int lineLength = 0;
    for (int i=0; i<xs.length; i++) {
        lineLength++;
        if (lineLength == w+1) {
            lineLength = 1;
            pw.println( );
        }
        pw.print(xs[i]+" ");
    }
    pw.close( );
}

```

```

public class Employee {

    private String name;
    private double hoursWorked; // hours worked this week
    private double hourlyRate = 30; // $'s per hour paid

    public Employee (String ident) {
        name = ident;
    }
    public void setHoursWorked (double worked) {
        hoursWorked = worked;
    }
    public double getRate ( ) {
        return hourlyRate;
    }
    public String toString ( ) {
        return name+"\n"
            +"hours worked this week: "+hoursWorked+"\n"
            +"hourly rate: $" +getRate()+"\n"
            +"earnings this week: $" +getRate()*hoursWorked;
    }
}

```

Question 2

A boss's hourly rate of pay is calculated at \$50 plus 50c for each level (1, 2, 3, ...) per hour.

A boss is paid a vehicle allowance of 40c per kilometre.

```

public class Boss extends Employee {
    private double allowancePerKilometre = 0.40;
    private int kilometresTravelled; //ks this week
    private double bossHourlyRate = 50;
    private int level;

    public Boss (String id, int theLevel) {
        // ...
    }
    public double getRate ( ) {
        // ...
    }
    public void setKilometres (int kilometres) {
        // ...
    }
    public String toString ( ) {
        // ...
    }
}

```

```

public Boss (String id, int theLevel) {
    super(id);
    level = theLevel;
}

public double getRate ( ) {
    return bossHourlyRate + 0.5*level;
}

public void setKilometres (int kilometres) {
    kilometresTravelled = kilometres;
}

public String toString ( ) {
    return super.toString( ) + "\n"
        +"kilometres travelled this week: "
        +kilometresTravelled
        +"\nvehicle allowance this week: "
        +0.4*kilometresTravelled;
}

```

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import genesis.*;
public class SqGUI extends JFrame {

    private JTextField number;

    public SqGUI ( ) {
        setTitle("Squaring");
        setBounds(400,150,200,130);
        number = new JTextField("0", 17);
        JButton squareRoot = new JButton("Square Root");
        Container c = getContentPane( );
        JPanel pMid = new JPanel( );
        pMid.add(number);
        c.add(pMid, "Center");
        JPanel pBot = new JPanel( );
        pBot.add(squareRoot);
        c.add(pBot, "South");
    }
}

```

Question 3(a)

Implement the code for the ‘Square Root’ action button.

Question 3(b)

Add the ‘Square’ action button and implement its code.

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

public class SqGUI extends JFrame {

    private JTextField number;

    public SqGUI ( ) {
        setTitle("Squaring");
        setBounds(400, 150, 200, 130);
        number = new JTextField("0", 17);
        JButton squareRoot =
            new JButton("Square Root");
```

```

squareRoot.addActionListener
    (new ActionListener( ) {
public void actionPerformed (ActionEvent e){
    double val =
        Double.parseDouble(number.getText( ));
    double root = Math.sqrt(val);
    number.setText(""+root);
    }
});
JButton square = new JButton("Square");
square.addActionListener(new ActionListener( ){
    public void actionPerformed (ActionEvent e){
        double val =
            Double.parseDouble(number.getText( ));
        double sq = val*val;
        number.setText(""+sq);
    }
});

```

```
Container c = getContentPane( );
JPanel pTop = new JPanel( );
pTop.add(square);
c.add(pTop, "North");
JPanel pMid = new JPanel( );
pMid.add(number);
c.add(pMid, "Center");
JPanel pBot = new JPanel( );
pBot.add(squareRoot);
c.add(pBot, "South");
```

```
}
}
```