

Week 12, Lecture 2

Working the Sample Exam



1

Announcements

- The **deadline for Labs 8, 9 and 10** is **today**.
- The **Prac Exam** will be held on Thursday and Friday **next week**. Make sure you know the details of your exam session; see the webpage.
- There will be **no lectures or lab tuts next week**.
- Your marked **Assignment 3** is on the web page .

2

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.**

3

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 $T \geq 50$.

4

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).

5

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.**



6

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.

7

even yet more Prac Exam

- You may bring to the exam the *Java Genesis* text book and any other written/printed material.
- **You cannot bring floppy disks** to the exam
- C-drive and G-drive will be as always, but there will be **no access to H-drive**.
- You will have access to the course web pages.
- **Make sure you bring your Student Card!**

8

Prac Exam results

- When all the Prac Exams have been graded, the mark for each student will be posted on the web page.
- We expect the grading to be completed by the end of swat-vac.
- A supplementary Prac Exam will be held, probably on Tues 9th. See web page for full details by end of swat-vac.

9

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.

10

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!**
- The exam is **NOT** open-book.

11

```
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);
    }
}
```

12

```

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
    // ...
}
}

```

13

```

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;
}

```

14

```

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( );
}

```

15

```

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;
    }
}

```

16

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.

17

```

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 ( ) {
        // ...
    }
}

```

18

```

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;
}

```

19

```

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");
    }
}

```

20

Question 3(a)

Implement the code for the 'Square Root' action button.

Question 3(b)

Add the 'Square' action button and implement its code.

21

```

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");
    }
}

```

22

```

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);
    }
});

```

23

```

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");
}
}

```

24