**Programming II – COMP 271 – John Mensing**

**9/12/2016 – week 1**

Website to check job/payment demand: <http://www.bls.gov/ooh/>

Check programming language desired by market: <http://www.tiobe.com/tiobe-index/>

Today: Review COMP 171

Objects & Class

Eclipse

String course = “Comp 271”

If(course == “Comp271”) 🡪 will give error, but will let compile it.

If(course.equals(“Comp271”)) 🡪 that’s the way to compare strings

Print the numbers from 1 to 10

int x=0

while (x<10)

{

X++;

S.O.P(x);

}

------------------------------------

x=0;

do

{

X++’

S.O.P(x);

} while (x<10);

--------------------------------------

int [ ] scores = new int [3];

double avgTest = findAvg(socres);

--------------------------------------

public static double findAvg(int[ ] nums)

{

double avg,sum;

for(int index=0, avg=0, sum=0; index < num.length; index++)

{

sum +=nums[index];

}

avg = sum/nums.length;

return avg;

}

------------------------------------------

Controlling the output – printing 2 decimals only:

System.out.print f(“%.2f” avgTest)

**Constructor has the same name as class**. No return type. Not even void.

Scanner indata new Scanner (System.in);

Indata.nextInt(0)

Sources 🡪 Generate Constructions using fields

Sources 🡪 Generate Getters and Setters

**Homework:**

**Review Comp 171**

**Look at Labs**

**Download Eclipse Neon**

**Design a class: List data and some methods (like rectangle class and its methods)**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**9/19/2016 – week 2**

Nice website to check it out: <http://www.eweek.com/mobile/>

Void/double/static – check out their rules

setWidth(width)

this.width

Width is an argument

This.width is an attribute

* Static methods – cannot access nonstatic methods
* Instance methods – can access Static variables and methods

**Be consistent**:

There is a length method like:

length( ) – string 🡪 this is not consistent, bc this should be an array

length – arrays

**Be complete**:

equals( )

equalsIgnoreCase( )

trim( )

format( ) – works similar to printf

Ex: System.out.printf(“Total is %.2f”, total);

Printf is common used to money

%f

String %s

Int - %d

**Use encapsulation**: data should be private

**Have a separate class**es: student / address

Wrappers (they took something simple and put other things around it):

Integer (class)– int

Double – double

Character – char

Interger.parseInt(“20”) 🡪 change a char to int

parseInt is a method of Class integer

----------------------------

driver.java – first version

**public** **class** driver {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println(" Number of objects " + CircleWithStaticVariableAndMethod.*getNumberOfObjects*());

// now create a circle

CircleWithStaticVariableAndMethod mycircle = **new** CircleWithStaticVariableAndMethod(2);

System.***out***.println(" Number of objects " + CircleWithStaticVariableAndMethod.*getNumberOfObjects*());

System.***out***.println(" Number of objects " + mycircle.*getNumberOfObjects*());

}

}

**myDate .java program combined with driver.java (check both files)**

Create a constructor

month day year

gets & sets for month, day, year

validation month & day

getMonthName

toString – Sep 19, 2016

To using eclipse to generate constructors go to menu Source/Generate Constructor using Fields

Mark 2nd box “Omit call of the Constructos…

Select month, day, year

After, go to menu Source/generate getters and setters  
Select month, day, year

-------------------------------------

Inheritance

Object – keeps date in hashcode

toString( )

myDate is a child of Object

**Homework: Lab1 – MyInteger**

Chapter 10 – page 400 #10.3

Skip parseInt

Instance can invoke static

Source/Generate Constructor using Fields

\*\*Prime number equal or greater the 3, is dividable by 1 and itself.

Week3

It’s inheritance because has the word **extend**

Example: (pizzadrv.java)

Pizza

Small-Pizza $8, .25 per toppins

Medium-Pizza $10, .50 per topping

Large-Pizza $12, .75 per topping

Pizza – we want to know: cost, size, order\_date, toppings

Tip: When typed protected Date order\_date (hover on Date and select java.util)

**!!! IMPORTANT**

Check it out later:

EXTENDS

SUPER

Test333.java

The first constructor buildt is C, b/c he’s the parent. B extends from C, and A extends from B. Extends means inheritance.

Super ( ) reach the first parent, there’s no way to reach 2 levels above.

Instace of check if a va is in that class.

Example: if (var instanceof Apple)

**Lab2 – Page 445 – #11.1**

Week 4 – 10/3/2016

Today

Review inheritance

Tee shirts

Abstract classes

Why use abstract classes

Revisit Pizza

Array must be a **double** or **int**

**\*\*See power point for more explanation about this week**

An abstract method cannot be contained in a nonabstract class. If a subclass of an abstract superclass does not implement all the abstract methods, the subclass must be defined abstract. In other words, in a nonabstract subclass extended from an abstract class, all the abstract methods must be implemented, even if they are not used in the subclass.

Knowing that, how to make pizzadrv.java better?

Make pizza abstract:

Pizza pz = new Pizza (“XLArge”) ;

We’ve seen up to now - Chap 12 and 13.

**\*\*HW:**

**Lab 2 has 2 parts: Triangle and Account – DUE TO THE NEXT CLASS**

In the 2nd part you should just write SavingsAccount (override); CheckingAccount (overdraft); withdraw method

You don’t need to write to much, only that and print methods to check answers.

Lab3 – use abstract geometric object.

The 2nd part will gonna talk next week.

Next week – 10/10 – Review interfaces, comparable, abstract. And we gonna start exception handling on chapter 12 – this isn’t going in the test.

**FIRST TEST IN 10/17 (Chapter 10, 11, 13)**

Week 5 – 10/10/2016

Lab3 –Pg. 540 #13.9

Use Circle from files for Lab3 – geometricObject

Comparable override equals method

**OVER THE WEEK : PREPARE FOR TEST, LAB1,2,3**

Review:

**Overloading**: Same name method but different parameters types or different number of paramenters.

Ex: sort (int[ ]) sort(double[ ]) Same sort name

**Overriding:** Child has the same name as the parent. Ex: toString ( ) toString( ) -- It’s important when we use compareTo

Test: See Review sheet

-----

Exceptions won’t be on the next test

See PPT Exceptions - on canvas

catch (exception type ex) }

code = process the exception

}

PrintWriter – Used to print a file. It’s valid for Scanner method, which does the same.

**Week6**

Employee – Abstract

Get\_paid()

Salary extends Employee

Override get\_paid()

Try {

//code

Integer.parseInt(“two”)

}

Catch(Exception ex) { 🡨 Put exception to check everything

// process the problem

}

Finally }

// all the time

}

Lab4 – page 490 #12.15

You can do in the 1 or 2 separated programs.

MyJavaFX

Create a project, select Build new Java Project, Use default JRE

NameofTheClass.Method(); 🡪 Application.launch(args);

Application is an abstract class. The abstract method is *start*. We have to override *start*.

Take a look into java docs fx: <https://docs.oracle.com/javafx/2/api/>

Awt

Swing

JavaFx 2.0 (Java 1.8)

MyJavaFX – 0,0 starts on the left up corner for squares

Circle starts in the middle and goes accordingly radius

ShowCircel.java

Creating a filled circle.

Color mycolor = new Color(1.0, 0.0, 1.0, .75);

Color.setFill(mycolor);

CircleCentered:

These 2 lines make the circle stays centered.

circle.centerXProperty().bind(pane.widthProperty().divide(2));

circle.centerYProperty().bind(pane.heightProperty().divide(2));

NodeStyleRotateDemo.java

Homework: Lab5 pg 578 #14/3 image/card

An example os using shuffle: page 438 – java.util.Collections.shuffle(ArrayList).

**Week 8**

Today: Events; Different ways to implement; Examples, Mouse, Ket Event

Layout mangers

Stack Pane

HBox

VBox

Grid Pane cols, row

Flow Pane

Border Pane

See PPT file on week8 folder. 🡪 explain what is events in java. Go over the PPT !!

5 different ways to do the same thing:

SimpleEventDemo outside

SimleEventDemoInner inside

SimpleEventDemo2 class implements

SimpleEventDemoAnonymousClass $1

SimpleEventDemoLambda e 🡪

Grab from Canvas – week8\_done – the professor’s nums.java

**Check it out**

ControlCircleWithoutEventHandling.java

MouseEventDemo.java

KeyEventDemo.java 🡪 check why uppercase isn’t working – compare to professor’s file

PathTransition.java 🡪 check/compare why mine the drawings are on the top left instead of centered.

HW: Lab6 is similar to Lab5

Need method: getChildren().clear()

STUDY!! NEXT TEST IS COMING! – close book, close notes.. similar to the labs

**TODAYS CLASS – week9 Nov 7th**

Revisit font

Review events

Lab7

Start GUI, Labels, Buttons, Checkboxes. RadioButtons, Text Fileds, Text areas

Test in 2 weeks - Nov 21th

Events – Register component to the event handler

Provide detail on how to handle the event

Check canvas for any missing labs

\*\*\*\*\*

Lab7 – see the file Lab7.docx in the week9 folder

Similar exercise in the book page 624 #15.19

**Lab 7 explained:**

Create instance of Pane

Create a circle 10 pixels – Math.random() – get width & getHeight of Pane

Create a color for fill

Add circle to Pane

SetOnMouseClicked

To move circle:

Set a new x

Set a new y

Set a new color

To prevent circle goes over the screen, we need to have:

X > (width – radius)

And

X < radius

Add one to counter

Else getCurrentTime

End time

Display “It took” + endTime-startTime “tons”

System.currentTimeMillis();

ButtonDemo.java

If you want to change line 17, you can add line 18 after word “Left”

There’s a tool tip method: SetToolTip(new ToolTip(“go to left”));

Take a look at line

Week 10 – 11/14/2016

Agenda:

More GUI components

List view

Slider

Video/Audio

Set up Lab8

Review for test

phonePad:

Text-Top (where is shown the number)

Numbers part (buttons) will be gridPane

The whole phonePad will be BorderPane. CellPane - center

When clicked on a button, take the label and put on text-top.

Review test – Chapters 12 (exceptions/files), 14(layout), 15 (events)

* Layout managers : Grid; Border; HBox; Vbox; StackPane; Flow
* Using text I/O: We can save data.
  + Example: Exception handling; Checked; Unchecked; Compiler requires; code to acknowledge.

Try {

// code

}

catch (IOException e)  
 {

// code for exception

}

* Exception handling: bc sometimes we can solve the problem, instead of catch the exception
  + Example: constructor with negative numbers: new Circle(-10); It’s better catch the exception to avoid that problem
* Event handler: 2 steps 🡪 1: list for event 2: register it

Open book portion:

Ex: What interfaces we need to implement

STUDY EXCEPTIONS!!

Try {

S1;

S2; - throws an exception here, goes to S4

S3;

}

catch (IOException e)  
 {

S4;

}

S5;

If you didn’t catch one exception, using Finally you guarantee catch S7 with code below;

Finally {

S7;

}

**Week11 – 11-21-2016**

Check out the week11 folder

Lab9 – use Loan.java to build Lab9

You can use for loop if you want to change interest, monthly payment

**Week12 11-28-2016**

Check out the week12 folder

**Binary**

|  |  |  |  |
| --- | --- | --- | --- |
| **File**  PrintWriter | **FileOutStream**  DataOutputStream | **File**  Scanner | **FileInputStream**  DataInputStream |
| Print()  Printf()  Println()  Close() | WriteChar()  WriteInt()  WriteDouble()  WriteUTF() | nextInt()  nextDouble()  next()  hasnext()  close() | readChar()  readInt()  readDouble()  readUTF(0  available() 🡪 o EOF Exception |

**Lab 10 pg 701 # 17.3 – Binary I/O**