CSE 205 Review Session #1

Questions: [kvlopez1@asu.edu](mailto:kvlopez1@asu.edu)

1. Concepts:

* Accessor/modifier(mutator) methods.
  + Public void setName()
  + Public int getCredits()
* Static modifier
  + Public Static void calculateTuition()
    - When calling the calculateTuition method:
    - Non static methods = Object.calculateTuition();
    - Static method = calculateTuitition();
* Constructors
  + Public Student()
  + When using the super() reference it should be the first thing in the constructor.
* Visibility modifiers (access specifiers)
  + private means visible inside this class only (including all its members);
  + protected — same as private + visible in subclasses too;
  + public — any client who sees the declaring class sees its public members.
* Method overloading
  + Rules:
    - The number of parameters is different for the methods.
    - The parameter types are different (like changing a parameter that was a float to an int).
* toString()

1. Abstract:

* An abstract class is a class that is declared abstract—it may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclassed.

Public abstract class Tuition

{

Public abstract void calculateTuition();

}

* An abstract method is a method that is declared without an implementation (without braces, and followed by a semicolon), like this:
* When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class. However, if it does not, then the subclass must also be declared abstract.

UMLs.

* Public (+)
* Private (-)
* Protected (#)
* Aggregation ◊
* Inheritance ∆

How to draw them?

|  |
| --- |
| **Class Name**  Student |
| **Class Variables**  - gpa: double  - name : String  - courses : ArrayList<Course> |
| 1. **Constructor** 2. **Methods**   +Student()  +Student(double, String, String)  +getName() : String  -getCredits() : int  +toString() : String |

How to implement it?

Public class Student

{

Private double gpa;

Private String name;

Private String lastName;

Private int credits;

Public Student() // Public Student (double gpa, String Name, String LastName)

{

this.gpa = gpa;

name = Name;

lastName = LastName;

credits = 18;

}

public String getName()

{

return name;

}

private int getCredits()

{

return credits;

}

public String toString()

{

String output;

Output = “\nName: \t\t” + name +

“\nCredits: \t\t” + credits;

return output;

}

}

Exercises

1. DecimalFormat dfmt = new DecimalFormat (“$#,###.00000”);

double test = 1234567.0898;

System.out.println(“number is: “ + dfmt.format(test));

Output:

$1,234,567.08980

1. String line = “[www.Netflix.com/watch/223489](http://www.Netflix.com/watch/223489)”;

Scanner scan = new Scanner(line).useDelimiter(“[/.]”);

While( scan.hasNext())

{

String output = scan.next();

System.out.println( output + “\t”);

}

output:

www

Netflix

com

watch

223489

1. Impot java.util.\*;

Public class ReviewSession

{

public static void main(String[] args)

{

ArrayList<String> majors = new ArrayList<String>();

majors.add(“Computer Science”);

majors.add(“Accounting”);

majors.add(0, “Chemistry”);

majors.add(“Physics”);

majors.set(2, “Art”);

majors.delete(3);

PrintArrayList(majors);

}

public static void PrintArrayList( ArrayList majors)

{

for (int I = 0; I < list.size(); I++)

System.out.print(majors.get(I) + “\t”);

}

}

Output:

Chemistry Computer Science Art

1. Public class Review

{

public static void main(String[] args)

{

String c1 = “Red”;

Color c2 = new Color(“Blue”);

System.out.println(“Before”);

System.out.println(“c1: ” + c1 + “\n” + “c2: “ + c2 + “\n”);

Swap(c1,c2);

System.out.println(“After”);

System.out.println(“c1: “ + c1 + “\n” + “c2: “ + c2 + “\n”);

}

public static void swap(String d1, Color d2)

{

d1 = “Yellow”;

d2.setValue(“Orange”);

System.out.println(“Middle”);

System.out.println(“d1: “ + d1 + “\n” + “d2: “ + d2 + “\n”);

}

private static class Color

{

private String color;

public Color( string Color)

{

color = Color;

}

public void setValue( String Color)

{

color = Color;

}

public String toString()

{

return color;

}

}

}//end class

1. Public String ReviewSessionPractice (int value1, char value2, double value3)

Signature?

ReviewSessionPractice(int, char, double)

Overloaded method?

Public String ReviewSessionPractice(double val1, char val2, String val3)