Name \_\_\_Eric Oliver\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (10 pts)Write the Java statement that will declare an array for the following table: (you do not need to assign the values – just the declaration statement).

|  |  |  |
| --- | --- | --- |
| 6.48 | 27.5 | 6.35 |
| 5.36 | 9.5 | 75 |
| 5 | 45 | 8.68 |
| 73.6 | 8.5 | 48.2 |

Double[][] tableOfNums;

1. (10 pts)Explain how the three visibility modifiers work.
   1. Protected

This modifier allows the member to only be accessed within its own package

* 1. Public

Means that all code can access this class, field or constructor

* 1. Private

This modifier only allows the member to be accessed in its own class.

1. (10 pts)Define/Contrast interfaces and abstract classes.

An abstract class can have abstract methods and non-abstract methods, while an interface can only have abstract methods. Abstract classes can provide the implementation of the interface and an interface cannot provide the implementation of an abstract class.

1. (10 pts)What is/are the benefits of using Generics? Why use Generics?

Generics are a way to define a class or method to be given types at compile time. We want the capabilities of specifying types in a method for a specific type.

This can be useful because we can overload them with different methods and call them.

1. (20 pts)Write a generic class below called **GenData**. It should contain a single variable called DATA. It should have a single constructor that takes an argument and stores it in DATA. It should also have the appropriate get and set methods for manipulating DATA.

public class GenData <DATA>{

private DATA d;

}

GenData(){}

GenData(DATA d){

This.d =DATA;

}

public DATA getD(){

return d;

}

Public DATA setD(DATA d){

this.d = d;

}

1. (15 pts)Given the class you implemented previously, in the main class below, write the code to create 2 objects of **GenData** type. Make an object store a String and the other a Double. (remember, generics don’t work with primitive types)

public class main {

public static void main(String[] args) {

GenData<String> data1 = new GenData<String>();

GenData<Double> data2 = new GenData<Double>();

}

}

1. (15 pts)Wiggles functions, are defined recursively as follows:

W(n) = 1 for n = 0

W(n) = 2 for n = 1

W(n) = W(n – 2) \* W(n - 1) \* n for n > 1

Write the Java recursive function for Wiggles.

W(int n){

If(n=0)

Return 1;

Else if (n = 1)

Return 2;

Else

Return W(n-2)\*W(n-1)\*n;

}

1. (10 pts)Suppose you have three classes; **Apple**, **Orange**, and **Banana**, and all three classes inherit from a class called **Fruit**. Given the main method below, write a SINGLE method called PrintType that can be used for all three statements. (Assume there is a toString() method for each class that just prints out the classes information. This should be what is printed in the PrintType method.)

public class example1

{

public static void main(String[] args)

{

//Print out the Fruit Information

print(new Orange());

print(new Apple());

print(new Banana());

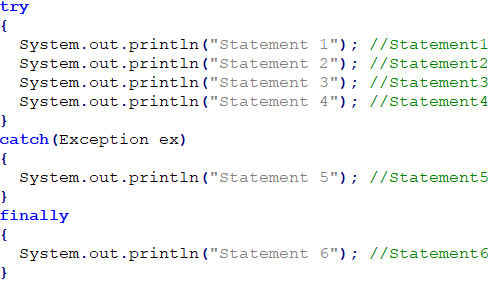
}

Public PrintType(fruit obj)

this.fruit.toString();

}

1. (10 pts)Consider the exception block below. Suppose that Statement3 has a problem and throws an exception. What is the output of the following block of code?



“Statement 5”

“Statement 6”

1. (15 pts)Answer the following questions concerning files.
   1. Give advantages of using binary files.

Binary IO does not require conversions

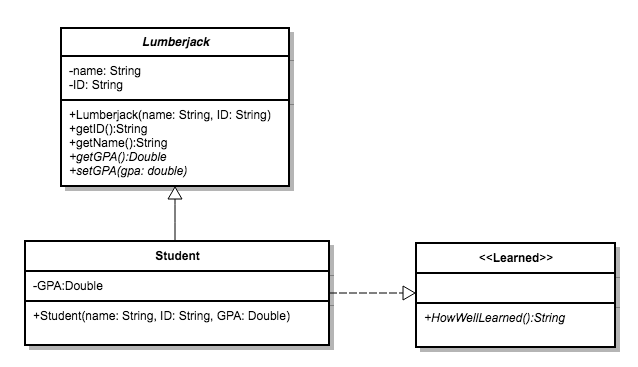
* 1. What is the purpose of the Serializable interface?

An object can implement the serializable interface to be written to a file

* 1. How is a Random Access file different? How does it read/write by?

A random-access file allows you to specify whether you want it to read or write. It does this byte by byte with the seek() method.

For Questions 11-13 consider the following UML Diagram.



1. (50pts)Implement the **Student** class below. The **HowWellLearned** method should just return a string indicating how good the student felt about his or her education.

Public class Student{

Private double GPA;

Student( String name, String ID, double GPA){

Super(name, ID);

This.GPA = GPA;

}

Public String HowWellLearned(){

Return “The student does not understand how the university expects them to absorb 15 weeks worth of knowledge in 6 weeks. There isn’t enough time in the day.”;

}

}

1. (25 pts)Consider the following Exception class. Rewrite your setGPA method below utilizing this exception class. If the GPA entered is not at least a 2.0, then this method should throw the exception. Otherwise it should set the GPA to the value given.

I’m pretty confused by this question. Why would we write a setGPA method in our exception class? Maybe I’m misinterpreting the question.

public class InvalidGPAException extends Exception

{

InvalidGPAException()

{

}

public string errorMessage()

{

return "Invalid GPA. Must be a minimum of 2.0";

}

}

setGPA(double GPA)throws InvalidGPAException

{

If(this.GPA >= 2.0)

This.GPA = this.GPA;

Else

Throw new InvalidGPAException(double GPA);

}

1. (25 pts)Consider the following block of code and Main Method. The current object has an invalid GPA being entered. Rewrite the main method that will handle the error caused in the code displaying the error message being thrown by the **Student** class.

public static void main(String[] args)

{

Student student1 = new Student("JimJim", "12345678", 0.0);

student1.setGPA(1.2);

}

public static void main(String[] args)

{

Try

{

Student student1 = new Student("JimJim", "12345678", 0.0);

student1.setGPA(1.2);

}

Catch(InvalidGPAException ex){

System.out.println(ex.errorMessage());

}

}

1. (10 pts)What are the commands in UNIX for the following tasks?
   1. To compile the file **finalExam.java**?

javac finalExam.java

* 1. To run the file **finalExam.java**?

java finalExam

* 1. To submit the file **tada.java** using the account csci230299?

submit csci230299 tada.java

1. (15 pts)Define the following terms:
   1. Abstraction

Only the essential details are displayed to the user

* 1. Encapsulation

Wrapping up of data under a single unit

* 1. Inheritance

Allows a child class to inherit certain variables and methods from the parent class

* 1. Polymorphism

The ability of an object to take many different forms. It allows us to perform the same action in many ways. Creating multiple objects with different variables.

* 1. Immutable

Once this object is created, we cannot change its content. Allows us to restrict the user from changing things we may not want changed.

Bonus: (5 pts)When talking about Multi-Threading, we discussed a few potential problems. One of them was something called Race Condition. What are Race Conditions in this context?

Multiple threads using the same variable can override that variable with the wrong information.