Charlson So

1. pg 696 algorithm workbench ex 3

**Class B**

public abstract class b{

private int m;

protected int n;

b(){

m=0;

n=0;

}

public void setM(int a){

}

public void setN(int a){

}

public int getM(){

return m;

}

public int getN(){

return n;

}

abstract double calc();

}

**Class D**

public class d extends b{

private double q;

protected double r;

d(){

q=0;

r=0;

}

public void setQ(double a){

q=a;

}

public void setR(double a){

r=a;

}

public double getQ(){

return q;

}

public double getR(){

return r;

}

@Override

public double calc(){

return q\*r;

}

}

1. pg 697 short answer 4

The reason why we have different control access is to determine the transparency in which certain data types are allowed to share and protect information. A protected member class will allow the subclass and the package to access the members. A private class will hide the variable from other classes within the package.

1. pg 697 short answer 7

Overloading is when two or more methods in one class have the same method name but different parameters.

Overriding is when two methods with the same name and parameters but one method is from the superclass and the other from the subclass. The subclass overrides the superclass. Overriding allows the subclass to provide specific programming for that method.

“Polymorphism is a type of overriding, not overloading. Overriding is a runtime concept while overloading is a compile time concept.”refer

1. pg 697 short answer 8

A java object can pass more than one is a test are considered polymorphic.

If “a” is a subclass of “b”, an object “a” can be created by reference to class “b”

a var = new b();

1. pg 702 ex 10

**ship.java**

public abstract class ship{

private String name;

private String yearBuilt;

ship(String n, String y){

name =n;

yearBuilt=y;

}

public void setName(String n){

name=n;

}

public void setYearBuilt(String y){

yearBuilt=y;

}

public String getName(){

return name;

}

public String getYearBuilt(){

return yearBuilt;

}

public String toString(){

return name+" "+yearBuilt;

}

}

CargoShip.java

public class CargoShip extends ship{

private int tonnage;

CargoShip(String n, String y, int t){

super(n,y);

tonnage =t;

}

public void setTonnage(int t){

tonnage =t;

}

public int getTonnage(){

return tonnage;

}

@Override

public String toString(){

return getName()+" Cargo Capacity: "+Integer.toString(tonnage);

}

}

CruiseShip.java

public class CruiseShip extends ship{

private int passengers;

CruiseShip(String n,String y, int p){

super(n,y);

passengers=p;

}

public void setPassengers(int p){

passengers=p;

}

public int getPassengers(){

return passengers;

}

@Override

public String toString(){

return getName()+"'s Number of Passengers"+Integer.toString(passengers);

}

}