**Find the Error:**

1. The error is that “expands” is not a keyword for a subclass. Using extend instead will fix the error.

Public class Car extends Vehicle

1. This error is caused because the constructor in the subclass does not use the super keyword to get the fields from the superclass. A constructor in the superclass is also missing (unless if it is implied that there is one).

Public Car(double c){

Super(c);

}

1. This error is caused by the subclass constructor missing its parameter for cost from the parent class. There is also a type error where the field cost which is type double is being assigned to the passenger field which is type int.

public Car(double c, int p)

{

Super(c);

passengers = p;

}

1. The errors in this problem are caused by the vehicle class not being an abstract class since Vehicle is meant to be the superclass. In the subclass Car the return type for getMilesPerGallon() method returns the wrong type.

Pubic abstract class Vehicle{

Public abstract double getMilesPerGallon();

}

Public class Car extends Vehicle{  
 private int mpg;

Public double getMilesPerGallon(){  
 return mpg;

}

}

**Algorithm Workbench:**

1. Public class Poodle extends Dog
2. Felis constructor than Tiger constructor

3.

B.java

public abstract class B {

    private int m;

    protected int n;

    public void setM(int m) {

        this.m = m;

    }

    public void setN(int n) {

        this.n = n;

    }

    public int getM() {

        return m;

    }

    public int getN() {

        return n;

    }

    public abstract double calc();

}

D.java

public class D extends B {

    private double q;

    protected double r;

    public void setQ(double q) {

        this.q = q;

    }

    public void setR(double r) {

        this.r = r;

    }

    public double getQ() {

        return q;

    }

    public double getR() {

        return r;

    }

    public double calc() {

        return r\*q;

    };

}