Figure Project 8.5

Overall Plan (Algorithm - step-by-step plan to make it happen):

1. The Figure class will have a center, draw, and erase method.

2. It will call the Triangle and Rectangle classes and will be overloaded by them, respectively, when they’re called

3. Displays the Triangle, Rectangle, and Figure class from the Demo class on the output

/\* Course CS112 Days & Time: September 10, 2015

\* Chapter Number: 8 Project Number: 5

\* Programmer: Broderick Higby

\* Program Title: Figure

\* Program Description: This program displays the different methods and demonstrates

\* the use of polymorphism, and late-binding.

\*/

public class Figure

{

String drawString;

String eraseString;

//This center method is called in the demo class

public void center()

{

draw();

erase();

System.out.println("This is the center method being called");

}

public void draw()

{

System.out.println("Draw Figure");

}

public void erase()

{

System.out.println("Erase Figure");

}

}

/\* Description: This is the child class, it calls from the base class "Figure"

\* and overloads the "erase" and "draw"

\* methods in the Figure class to output a triangle.

\*

\*/

public class Triangle extends Figure

{

public void draw() //draw method overrides Figure classes method

{

System.out.println("Draw Triangle");

}

public void erase() //erase method overrides Figure classes method

{

System.out.println("Erase Triangle");

}

}

/\* Description: This is the child class, it calls from the base class

\* "Figure" and overloads the "erase" and "draw"

\* methods in the Figure class to output a rectangle.

\*

\*/

public class Rectangle extends Figure

{

public void draw() //draw method is called from "Figure"

{ //draw method overrides Figure classes method

System.out.println("Draw Rectangle");

}

public void erase()

{ //erase method overrides Figure classes method

System.out.println("Erase Rectangle");

}

}

/\* Description: This is the demo class for the Figure program, it uses

\* late binding to call the center method from Figure,

\* Triangle, and Rectangle classes override the Figure method

\*/

public class FigureDemo

{

public static void main(String[] args)

{

//These constructions are used to call the center method

Figure demoFigure = new Figure();

Figure demoTriangle = new Triangle();

Figure demoRectangle = new Rectangle();

//The demo classes call the methods' "center" from "Figure" constructed above, then it goes over to the other classes

//and pulls the methods from Triangle and Rectangle. Those override Figure's methods

demoFigure.center();

demoTriangle.center();

demoRectangle.center();

}

}