



**BSC. (HONS) IN ELECTRONICS AND TELECOMMUNICATIONS  
ENGINEERING**

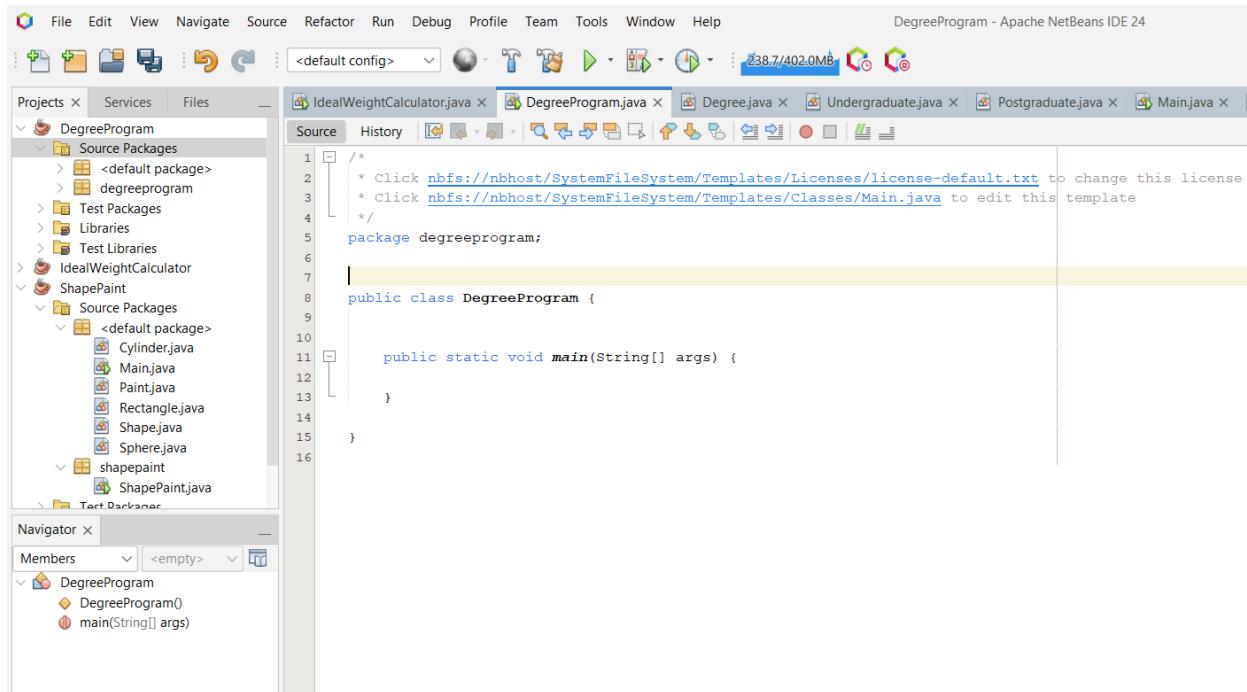
## **ECS2301-Software Engineering and Project**

**LAB ASSIGNMENT NO. : 04**

**INDEX NUMBER** : 23UG1- 0152\_Akindu Randira

15<sup>th</sup> FEBRUARY 2025

1. Create a class 'Degree' having a method 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the method by creating an object of each of the three classes.

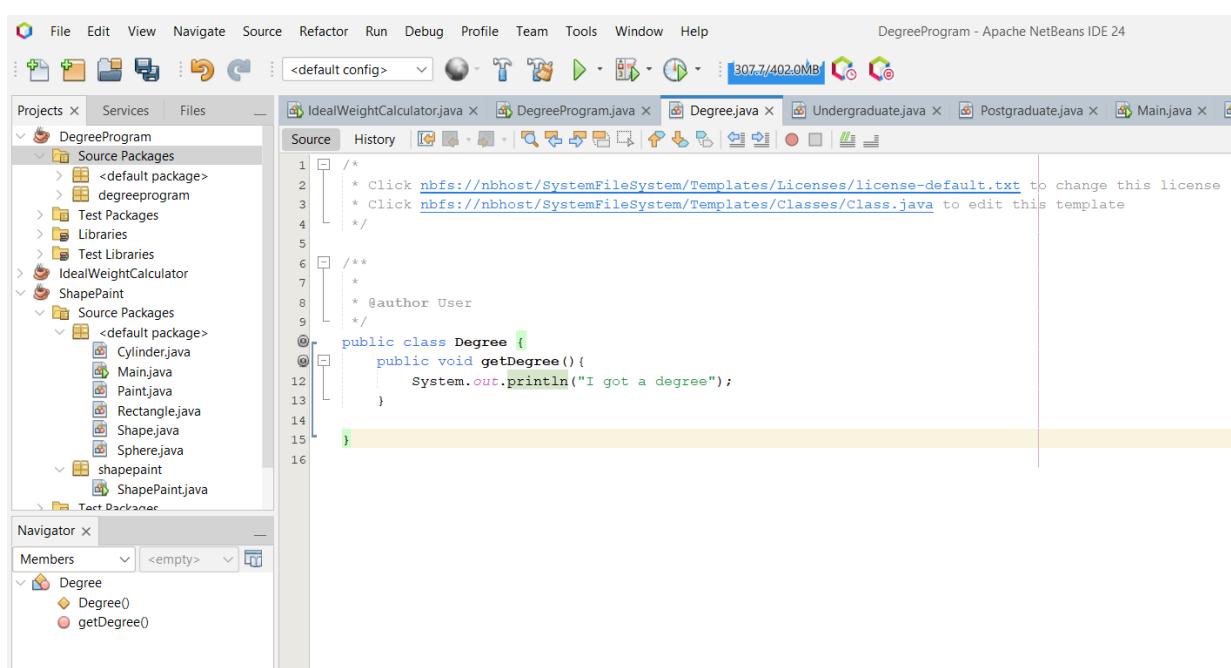


```

 1  /*
 2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 4  */
 5  package degreeprogram;
 6
 7  public class DegreeProgram {
 8
 9      public static void main(String[] args) {
10
11      }
12
13  }
14
15
16

```

Figure 1: DegreeProgram Class



```

 1  /*
 2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 4  */
 5
 6  /**
 7  *
 8  * @author User
 9  */
10 public class Degree {
11     public void getDegree() {
12         System.out.println("I got a degree");
13     }
14
15
16

```

Figure 2 : Degree Class

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard NetBeans icons.
- Project Explorer:** Shows the project structure under "DegreeProgram".
- Source Editor:** Displays the code for the `Undergraduate.java` file. The code is as follows:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author User
 */
public class Undergraduate extends Degree {
    public void getDegree() {
        System.out.println("I am an Undergraduate");
    }
}
```

- Navigator:** Shows the members of the `Undergraduate` class.

Figure 3 : Undergraduate Class

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Menu Bar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Toolbar:** Standard NetBeans icons.
- Project Explorer:** Shows the project structure under "DegreeProgram".
- Source Editor:** Displays the code for the `Postgraduate.java` file. The code is as follows:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author User
 */
public class Postgraduate extends Degree {
    public void getDegree() {
        System.out.println("I am a Postgraduate");
    }
}
```

- Navigator:** Shows the members of the `Postgraduate` class.

Figure 4 : Postgraduate Class

The screenshot shows the Apache NetBeans IDE 24 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar indicates "DegreeProgram - Apache NetBeans IDE 24". The Projects tab is selected, showing a tree view of packages: DegreeProgram (Source Packages, Test Packages, Libraries, Test Libraries), IdealWeightCalculator, and ShapePaint (Source Packages, Test Packages). The Source tab is active, displaying the Main.java code. The code defines a public class Main with a main method. Inside the main method, objects of Degree, Undergraduate, and Postgraduate classes are created and their getDegree methods are called. The Navigator tab shows the members of the Main class: Main0 and main(String[] args). The status bar at the bottom shows memory usage: 173.2/402.0MB.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
public class Main {
    public static void main(String[] args) {
        Degree degree = new Degree();
        degree.getDegree(); // Outputs: I got a degree

        Undergraduate ug = new Undergraduate();
        ug.getDegree(); // Outputs: I am an Undergraduate

        Postgraduate pg = new Postgraduate();
        pg.getDegree(); // Outputs: I am a Postgraduate
    }
}
```

Figure 5 : Main Class

This screenshot shows the Apache NetBeans IDE 24 interface with the same project structure as Figure 5. The Main.java code is identical. In the bottom panel, the Output window titled "Output - DegreeProgram (run)" displays the program's output: "I got a degree", "I am an Undergraduate", and "I am a Postgraduate", followed by "BUILD SUCCESSFUL (total time: 0 seconds)". The status bar at the bottom shows memory usage: 244.8/402.0MB.

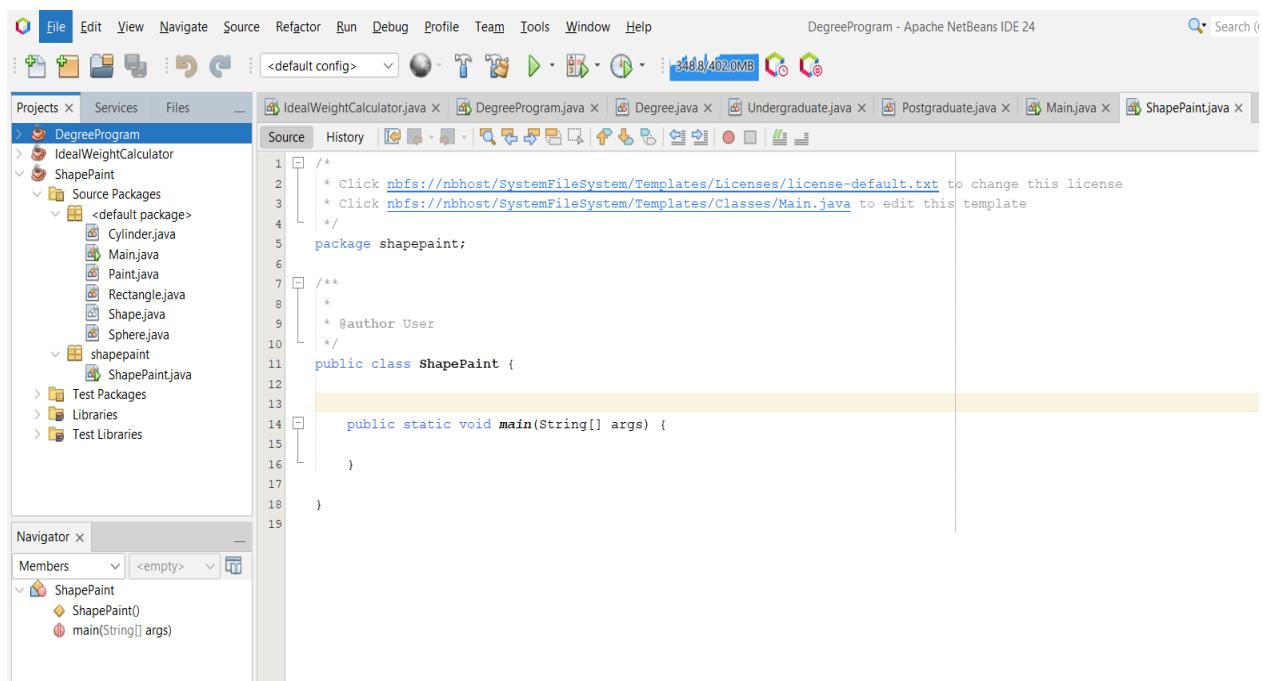
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
public class Main {
    public static void main(String[] args) {
        Degree degree = new Degree();
        degree.getDegree(); // Outputs: I got a degree

        Undergraduate ug = new Undergraduate();
        ug.getDegree(); // Outputs: I am an Undergraduate

        Postgraduate pg = new Postgraduate();
        pg.getDegree(); // Outputs: I am a Postgraduate
    }
}
```

Figure 6 : Output

2. Develop a class hierarchy of shapes and write a program that computes the amount of paint needed to paint different objects. The hierarchy will consist of a parent class Shape with three derived classes - Sphere, Rectangle, and Cylinder. For the purposes of this exercise, the only attribute a shape will have is a name and the method of interest will be one that computes the area of the shape (surface area in the case of three-dimensional shapes). Do the following.
- Write an abstract class Shape with the following properties:
    - An instance variable shapeName of type String
    - An abstract method area()
    - A toString method that returns the name of the shape



The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar indicates "DegreeProgram - Apache NetBeans IDE 24". The toolbar has various icons for file operations like Open, Save, and Build. The Projects tab shows a project named "DegreeProgram" with several Java files: IdealWeightCalculator.java, DegreeProgram.java, Degree.java, Undergraduate.java, Postgraduate.java, Main.java, and ShapePaint.java. The Files tab shows the same files. The Source tab is active, displaying the content of ShapePaint.java:

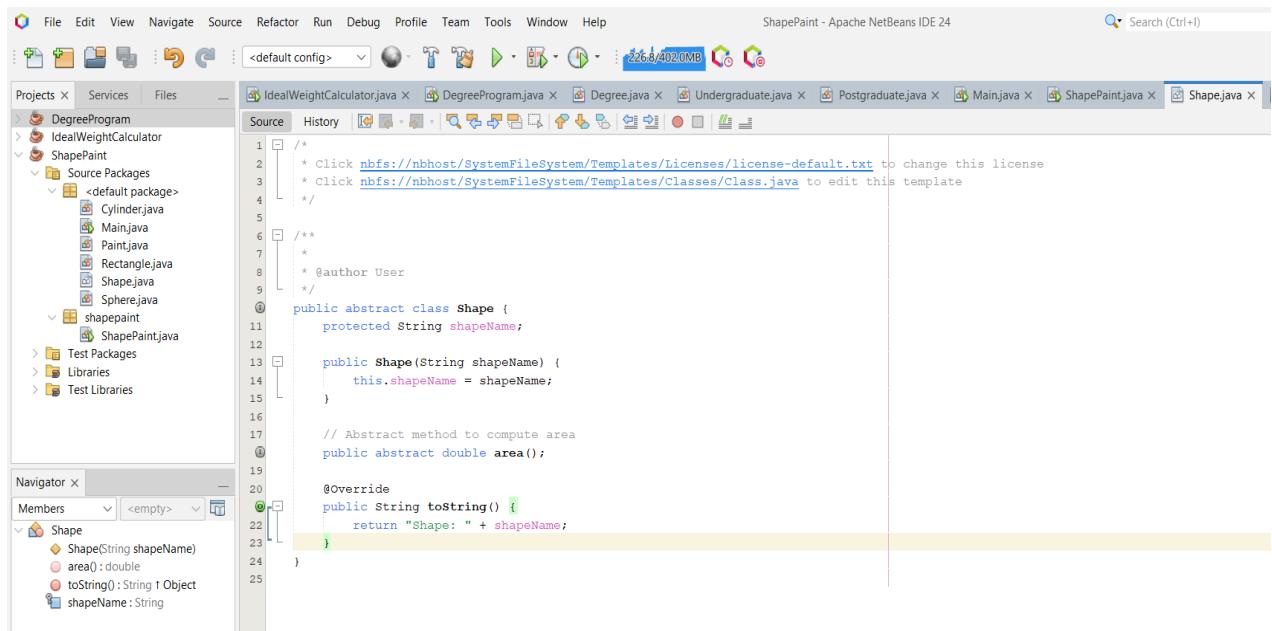
```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5   package shapepaint;
6
7  /**
8   *
9   * @author User
10  */
11 public class ShapePaint {
12
13     public static void main(String[] args) {
14
15     }
16
17 }
18
19

```

The Navigator tab shows the members of the ShapePaint class: ShapePaint and main(String[] args).

Figure 7: ShapePaint Class



The screenshot shows the Apache NetBeans IDE 24 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "ShapePaint - Apache NetBeans IDE 24". The toolbar has various icons for file operations like Open, Save, and Build. The Projects tab in the left sidebar lists "DegreeProgram", "IdealWeightCalculator", and "ShapePaint" (selected). The Files tab shows files like "IdealWeightCalculator.java", "DegreeProgram.java", "Degree.java", "Undergraduate.java", "Postgraduate.java", "Main.java", "ShapePaint.java", and "Shape.java". The Source tab displays the code for the "Shape" class:

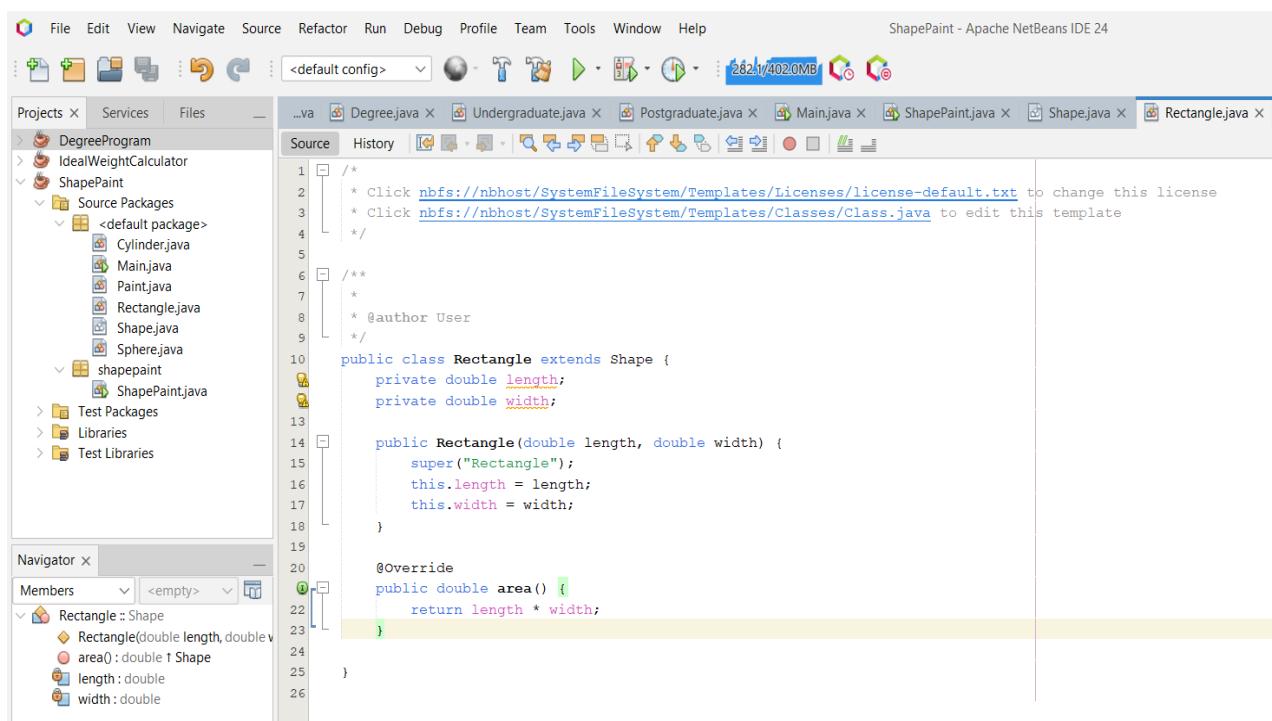
```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6 /**
7 *
8 * @author User
9 */
10 public abstract class Shape {
11     protected String shapeName;
12
13     public Shape(String shapeName) {
14         this.shapeName = shapeName;
15     }
16
17     // Abstract method to compute area
18     public abstract double area();
19
20     @Override
21     public String toString() {
22         return "Shape: " + shapeName;
23     }
24 }
25

```

The Navigator tab shows members of the "Shape" class: "Shape(String shapeName)", "area():double", "toString():String", and "shapeName : String".

Figure 8 : Shape Class



The screenshot shows the Apache NetBeans IDE 24 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "ShapePaint - Apache NetBeans IDE 24". The toolbar has various icons for file operations like Open, Save, and Build. The Projects tab in the left sidebar lists "DegreeProgram", "IdealWeightCalculator", and "ShapePaint" (selected). The Files tab shows files like "IdealWeightCalculator.java", "DegreeProgram.java", "Degree.java", "Undergraduate.java", "Postgraduate.java", "Main.java", "ShapePaint.java", "Shape.java", and "Rectangle.java". The Source tab displays the code for the "Rectangle" class:

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6 /**
7 *
8 * @author User
9 */
10 public class Rectangle extends Shape {
11     private double length;
12     private double width;
13
14     public Rectangle(double length, double width) {
15         super("Rectangle");
16         this.length = length;
17         this.width = width;
18     }
19
20     @Override
21     public double area() {
22         return length * width;
23     }
24 }
25

```

The Navigator tab shows members of the "Rectangle" class: "Rectangle(double length, double width)", "area():double", "length : double", and "width : double".

Figure 9 : Rectangle Class

The screenshot shows the Apache NetBeans IDE 24 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "ShapePaint - Apache NetBeans IDE 24". The toolbar has various icons for file operations. The Projects tab is selected, showing a tree view of projects: DegreeProgram, IdealWeightCalculator, ShapePaint (with Source Packages, Test Packages, Libraries, and Test Libraries), and Main.java. The Files tab is also visible. The Source tab is active, displaying the code for the `Sphere` class. The code defines a class `Sphere` that extends `Shape`. It has a private double variable `radius` and a constructor that takes a double radius. It overrides the `area()` method to calculate the surface area of a sphere. The Navigator tab shows the members of the `Sphere` class: `Sphere(double radius)`, `area(): double`, and `radius: double`.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
/**
 *
 * @author User
 */
public class Sphere extends Shape {
    private double radius;
    public Sphere(double radius) {
        super("Sphere");
        this.radius = radius;
    }
    @Override
    public double area() {
        // Surface area of a sphere = 4 * PI * r^2
        return 4 * Math.PI * radius * radius;
    }
}
```

Figure 10 : Sphere Class

The screenshot shows the Apache NetBeans IDE 24 interface, similar to Figure 10. The Projects tab is selected, showing the same project structure. The Source tab is active, displaying the code for the `Cylinder` class. The code defines a class `Cylinder` that extends `Shape`. It has private double variables `radius` and `height` and a constructor that takes both. It overrides the `area()` method to calculate the surface area of a cylinder. The Navigator tab shows the members of the `Cylinder` class: `Cylinder(double radius, double height)`, `area(): double`, `height: double`, and `radius: double`.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
/**
 *
 * @author User
 */
public class Cylinder extends Shape {
    private double radius;
    private double height;
    public Cylinder(double radius, double height) {
        super("Cylinder");
        this.radius = radius;
        this.height = height;
    }
    @Override
    public double area() {
        // Surface area of a cylinder = 2nr(r + h)
        return 2 * Math.PI * radius * (radius + height);
    }
}
```

Figure 11 : Cylinder Class

The screenshot shows the Apache NetBeans IDE 24 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "ShapePaint - Apache NetBeans IDE 24". The toolbar has icons for file operations like Open, Save, and Build. The Projects tab shows a project structure with DegreeProgram, IdealWeightCalculator, and ShapePaint. The ShapePaint package contains Source Packages (Cylinder.java, Main.java, Paint.java, Rectangle.java, Shape.java, Sphere.java) and shapepaint (ShapePaint.java). The Files tab shows files like Undergraduate.java, Postgraduate.java, Main.java, ShapePaint.java, Shape.java, Rectangle.java, Sphere.java, Cylinder.java, and Paint.java. The Source tab displays the Paint.java code:

```

1 /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5
6 /**
7  *
8  * @author User
9  */
10 public class Paint {
11     private double coverage; // e.g., how many square meters 1 liter can cover
12
13     public Paint(double coverage) {
14         this.coverage = coverage;
15     }
16
17     public double amountNeeded(Shape shape) {
18         return shape.area() / coverage;
19     }
20 }

```

The Navigator tab shows members of the Paint class: Paint(double coverage), amountNeeded(Shape shape): dc, and coverage : double.

Figure 12 : Paint Class

The screenshot shows the Apache NetBeans IDE 24 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "ShapePaint - Apache NetBeans IDE 24". The toolbar has icons for file operations like Open, Save, and Build. The Projects tab shows a project structure with DegreeProgram, IdealWeightCalculator, and ShapePaint. The ShapePaint package contains Source Packages (Cylinder.java, Main.java, Paint.java, Rectangle.java, Shape.java, Sphere.java) and shapepaint (ShapePaint.java). The Files tab shows files like Undergraduate.java, Postgraduate.java, Main.java, ShapePaint.java, Shape.java, Rectangle.java, Sphere.java, Cylinder.java, and Paint.java. The Source tab displays the Main.java code:

```

1 /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5
6 /**
7  *
8  * @author User
9  */
10 public class Main {
11     public static void main(String[] args) {
12         // Let's assume 1 liter of paint covers 10 square meters
13         Paint paint = new Paint(10);
14
15         // Create shape objects
16         Shape rectangle = new Rectangle(5, 10);
17         Shape sphere = new Sphere(7);
18         Shape cylinder = new Cylinder(4, 10);
19
20         // Calculate and display amount of paint needed
21         System.out.println(rectangle + " needs " + paint.amountNeeded(rectangle) + " liters of paint.");
22         System.out.println(sphere + " needs " + paint.amountNeeded(sphere) + " liters of paint.");
23         System.out.println(cylinder + " needs " + paint.amountNeeded(cylinder) + " liters of paint.");
24     }
25 }

```

The Navigator tab shows members of the Main class: Main(), Main0, and main(String[] args).

Figure 13 : Main Class

The screenshot shows the Apache NetBeans IDE 24 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar reads "ShapePaint - Apache NetBeans IDE 24". The toolbar contains various icons for file operations like Open, Save, and Build.

The Projects tab shows a hierarchy of projects: DegreeProgram, IdealWeightCalculator, ShapePaint (selected), Source Packages, Test Packages, Libraries, and Test Libraries. Under ShapePaint, there are Source Packages (Cylinder.java, Main.java, Paint.java, Rectangle.java, Shape.java, Sphere.java) and shapepaint (ShapePaint.java).

The Source tab displays the Main.java code:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
/**
 *
 * @author User
 */
public class Main {
    public static void main(String[] args) {
        // Let's assume 1 liter of paint covers 10 square meters
        Paint paint = new Paint(10);

        // Create shape objects
        Shape rectangle = new Rectangle(5, 10);
        Shape sphere = new Sphere(7);
        Shape cylinder = new Cylinder(4, 10);

        // Calculate and display amount of paint needed
        System.out.println(rectangle + " needs " + paint.amountNeeded(rectangle) + " liters of paint.");
        System.out.println(sphere + " needs " + paint.amountNeeded(sphere) + " liters of paint.");
        System.out.println(cylinder + " needs " + paint.amountNeeded(cylinder) + " liters of paint.");
    }
}
```

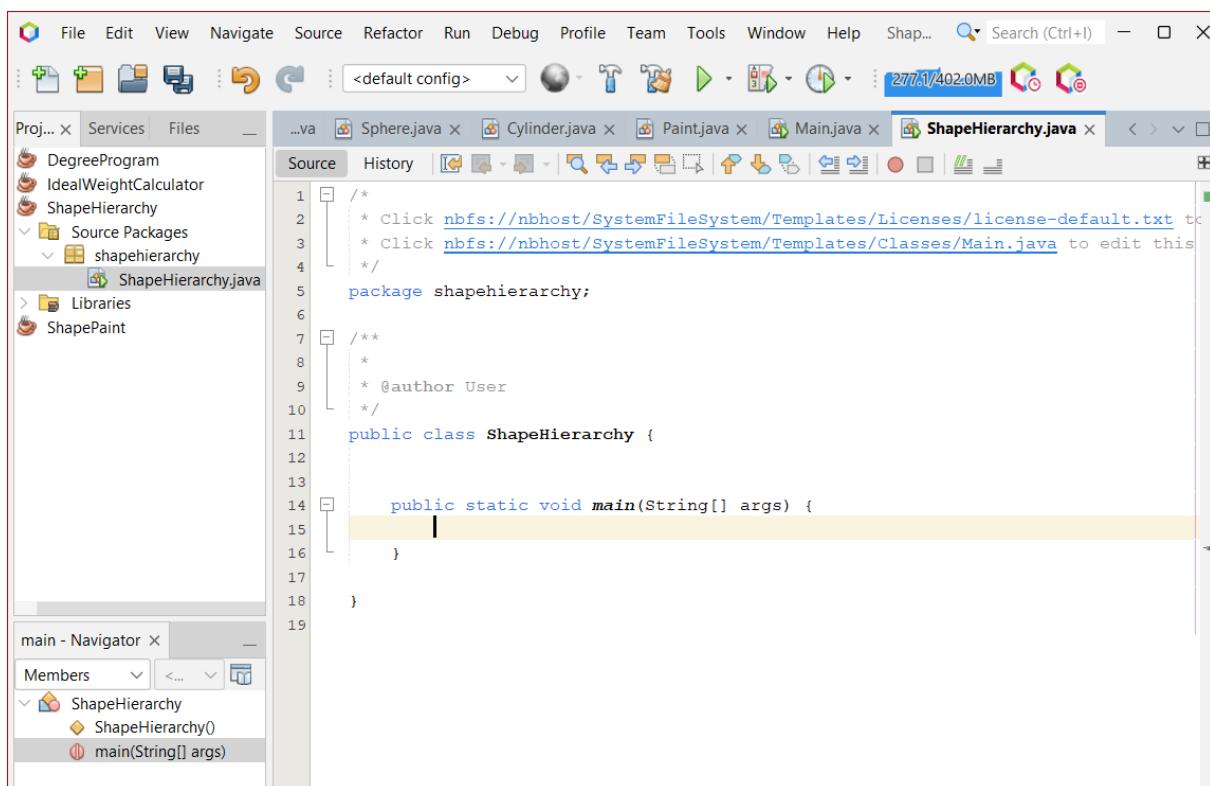
The Navigator tab shows Members: Main, Main0, and main(String[] args).

The Output tab shows the run results:

```
run:
Shape: Rectangle needs 5.0 liters of paint.
Shape: Sphere needs 61.57521601035994 liters of paint.
Shape: Cylinder needs 35.18583772020568 liters of paint.
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 14 : Output

- b) The file Sphere.java contains a class for a sphere which is a descendant of Shape. A sphere has a radius and its area (surface area) is given by the formula  $4\pi r^2$ . Define similar classes for a rectangle and a cylinder. Both the Rectangle class and the Cylinder class are descendants of the Shape class. A rectangle is defined by its length and width and its area is length times width. A cylinder is defined by a radius and height and its area (surface area) is  $\pi r^2 h$ . Define the `toString` method in a way similar to that for the Sphere class.



The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer (Left):** Shows files like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, and ShapePaint. A red box highlights the ShapeHierarchy.java file under the shapehierarchy package.
- Code Editor (Center):** Displays the `ShapeHierarchy.java` source code. The code defines a public class `ShapeHierarchy` with a main method. The `main` method is currently empty, indicated by a yellow highlighting bar under the cursor.
- Navigator (Bottom Left):** Shows the members of the `ShapeHierarchy` class, including the constructor `ShapeHierarchy()` and the `main` method.
- Toolbar and Status Bar (Top):** Standard NetBeans toolbar and status bar showing memory usage (277.1/402.0MB).

```

1 /**
2 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this
4 */
5 package shapehierarchy;
6
7 /**
8 *
9 * @author User
10 */
11 public class ShapeHierarchy {
12
13
14     public static void main(String[] args) {
15
16     }
17
18 }
19

```

Figure 15 : ShapeHierarchy Class

The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer:** Shows files like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages, Main.java, ShapeHierarchy.java, and Shape.java.
- Code Editor:** Displays the `Shape.java` file content:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
4  */
5
6 /**
7 *
8 * @author User
9 */
10 public abstract class Shape {
11     protected String shapeName;
12
13     public Shape(String shapeName) {
14         this.shapeName = shapeName;
15     }
16
17     // Abstract method to compute area
18     public abstract double area();
19
20     @Override
21     public String toString() {
22         return "Shape: " + shapeName;
23     }
24 }
```
- Navigator:** Shows the members of the `Shape` class, including `area()` and `toString()`.

Figure 16 : Shape Class

The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer:** Shows files like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages, Main.java, ShapeHierarchy.java, Shape.java, and Sphere.java.
- Code Editor:** Displays the `Sphere.java` file content:

```
1 /*
2 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
3 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
4 */
5
6 /**
7 *
8 * @author User
9 */
10 public class Sphere extends Shape {
11     private double radius;
12
13     public Sphere(double radius) {
14         super("Sphere");
15         this.radius = radius;
16     }
17
18     @Override
19     public double area() {
20         // Surface area of a sphere = 4 * PI * r^2
21         return 4 * Math.PI * radius * radius;
22     }
23
24     @Override
25     public String toString() {
26         return super.toString() + " with radius " + radius;
27     }
28 }
```
- Navigator:** Shows the members of the `Sphere` class, including `area()` and `toString()`.

Figure 17 : Sphere Class

The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer (Left):** Shows the project structure with packages like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages, Test Packages, Libraries, Test Libraries, and ShapePaint.
- Code Editor (Center):** Displays the `Rectangle.java` file content. The code defines a class `Rectangle` that extends `Shape`. It has private fields `length` and `width`, a constructor that takes `length` and `width`, and overrides `area()` and `toString()`.
- Navigator (Bottom Left):** Shows the members of the `Rectangle` class, including its constructor, `area()`, and `toString()` methods, along with `length` and `width` fields.

```

1 /**
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license.
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4 */
5
6 /**
7 *
8 * @author User
9 */
10 public class Rectangle extends Shape {
11     private double length;
12     private double width;
13
14     public Rectangle(double length, double width) {
15         super("Rectangle");
16         this.length = length;
17         this.width = width;
18     }
19
20     @Override
21     public double area() {
22         return length * width;
23     }
24
25     @Override
26     public String toString() {
27         return super.toString() + " with length " + length + " and width " + width;
28     }
29
30 }

```

Figure 18 : Rectangle Class

The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer (Left):** Shows the project structure with packages like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages, Test Packages, Libraries, Test Libraries, and ShapePaint.
- Code Editor (Center):** Displays the `Cylinder.java` file content. The code defines a class `Cylinder` that extends `Shape`. It has private fields `radius` and `height`, a constructor that takes `radius` and `height`, and overrides `area()` and `toString()`.
- Navigator (Bottom Left):** Shows the members of the `Cylinder` class, including its constructor, `area()`, and `toString()` methods, along with `radius` and `height` fields.

```

1 /**
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license.
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4 */
5
6 /**
7 *
8 * @author User
9 */
10 public class Cylinder extends Shape{
11     private double radius;
12     private double height;
13
14     public Cylinder(double radius, double height) {
15         super("Cylinder");
16         this.radius = radius;
17         this.height = height;
18     }
19
20     @Override
21     public double area() {
22         // Surface area of a cylinder = 2nr(r + h)
23         return 2 * Math.PI * radius * (radius + height);
24     }
25
26     @Override
27     public String toString() {
28         return super.toString() + " with radius " + radius + " and height " + height;
29     }
30
31 }

```

Figure 19 : Cylinder Class

The screenshot shows the NetBeans IDE interface. The Project Explorer on the left lists several Java files: DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages (<default package> containing Cylinder.java, Main.java, Rectangle.java, Shape.java), Test Packages, Libraries, Test Libraries, and ShapePaint. The Navigator pane shows members of the Main class: Main, Main(), and main(String[] args). The Source editor on the right displays the Main.java code:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5
6 /**
7 *
8 * @author User
9 */
10 public class Main {
11     public static void main(String[] args) {
12         // Create shape objects
13         Shape sphere = new Sphere(7);
14         Shape rectangle = new Rectangle(5, 10);
15         Shape cylinder = new Cylinder(4, 10);
16
17         // Display shape information and their area
18         System.out.println(sphere + " has an area of " + sphere.area());
19         System.out.println(rectangle + " has an area of " + rectangle.area());
20         System.out.println(cylinder + " has an area of " + cylinder.area());
21     }
22 }
```

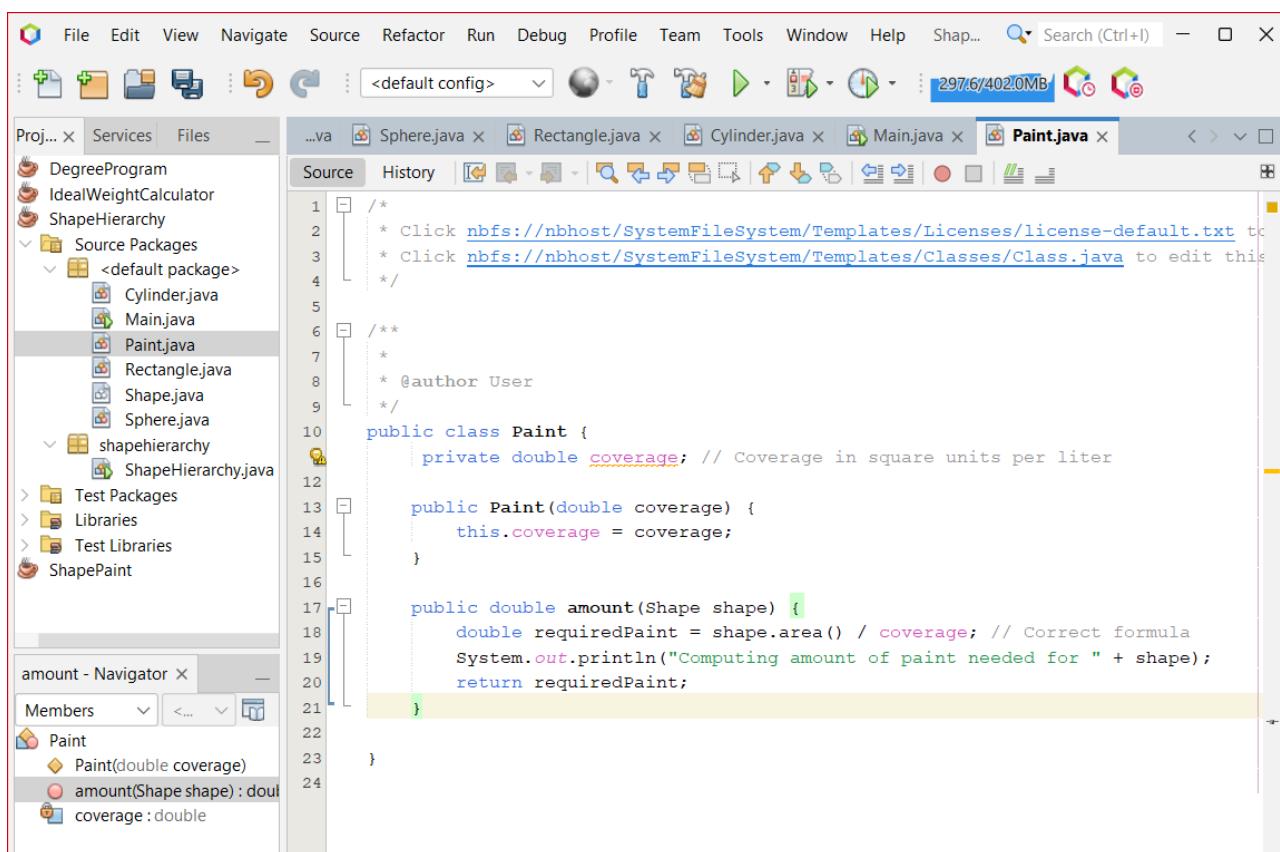
Figure 20 : Main Class

The screenshot shows the NetBeans IDE interface with a red box highlighting the Main.java file in the Project Explorer. The Source editor displays the same Main.java code as in Figure 20. Below the editor, the Output window titled "Output - ShapeHierarchy (run)" shows the execution results:

```
run:
Shape: Sphere with radius 7.0 has an area of 615.7521601035994
Shape: Rectangle with length 5.0 and width 10.0 has an area of 50.0
Shape: Cylinder with radius 4.0 and height 10.0 has an area of 351.85837720205683
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 21 : Output

- c) The file Paint.java contains a class for a type of paint (which has a "coverage" and a method to compute the amount of paint needed to paint a shape). Correct the return statement in the amount method so the correct amount will be returned. Use the fact that the amount of paint needed is the area of the shape divided by the coverage for the paint. (NOTE: Leave the print statement - it is there for illustration purposes, so you can see the method operating on different types of Shape objects.)



```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
 * see the conditions of distribution.  A copy of the license
 * should be located in the root of this project in a file named
 * license.txt
 */
/**
 *
 * @author User
 */
public class Paint {
    private double coverage; // Coverage in square units per liter

    public Paint(double coverage) {
        this.coverage = coverage;
    }

    public double amount(Shape shape) {
        double requiredPaint = shape.area() / coverage; // Correct formula
        System.out.println("Computing amount of paint needed for " + shape);
        return requiredPaint;
    }
}

```

Figure 22 : Paint Class

The screenshot shows the Apache NetBeans IDE interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and Search (Ctrl+). The title bar says "ShapeHierarchy - Apache NetBeans IDE 24". The left sidebar displays the project structure under "Proj... X Services Files". The main editor window shows the "Main.java" file with the following code:

```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5
6  /**
7   *
8   * @author User
9   */
10 public class Main {
11     public static void main(String[] args) {
12         // Define paint with a coverage of 350 square units per liter
13         Paint paint = new Paint(350);
14
15         // Create shape objects
16         Shape sphere = new Sphere(7);
17         Shape rectangle = new Rectangle(5, 10);
18         Shape cylinder = new Cylinder(4, 10);
19
20         // Display shape information and their area
21         System.out.println(sphere + " has an area of " + sphere.area());
22         System.out.println(rectangle + " has an area of " + rectangle.area());
23         System.out.println(cylinder + " has an area of " + cylinder.area());
24
25         // Compute and display paint needed
26         System.out.println("Paint needed for " + sphere + ": " + paint.amount(sphere) + " liters.");
27         System.out.println("Paint needed for " + rectangle + ": " + paint.amount(rectangle) + " liters.");
28         System.out.println("Paint needed for " + cylinder + ": " + paint.amount(cylinder) + " liters.");
29     }
30 }

```

Figure 23 : Main Class

The screenshot shows the Apache NetBeans IDE interface with the "Output - ShapeHierarchy (run)" window open at the bottom. The output window displays the following run log:

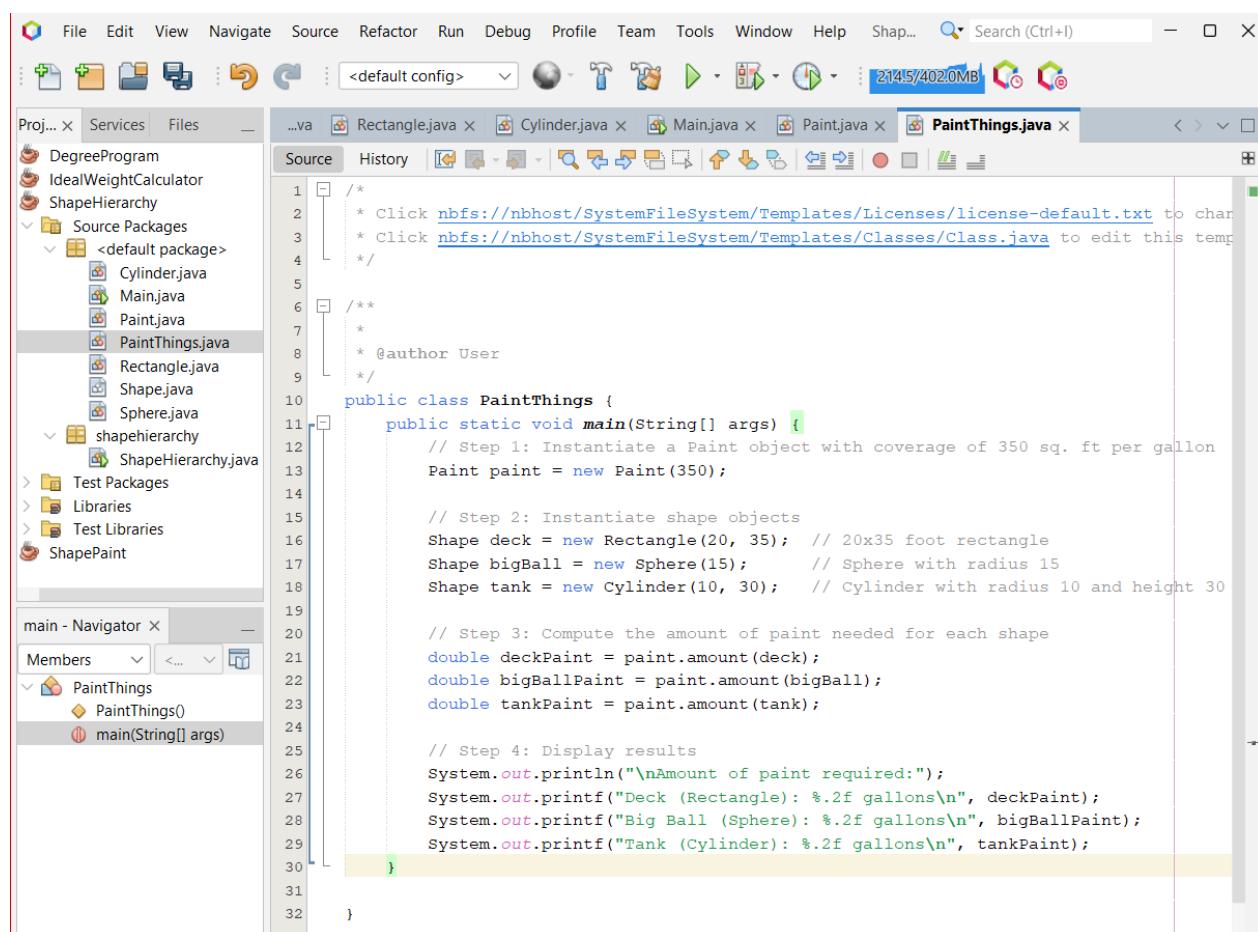
```

run:
Shape: Sphere with radius 7.0 has an area of 615.7521601035994
Shape: Rectangle with length 5.0 and width 10.0 has an area of 50.0
Shape: Cylinder with radius 4.0 and height 10.0 has an area of 351.85837720205683
Computing amount of paint needed for Shape: Sphere with radius 7.0
Paint needed for Shape: Sphere with radius 7.0: 1.759291886010284 liters.
Computing amount of paint needed for Shape: Rectangle with length 5.0 and width 10.0
Paint needed for Shape: Rectangle with length 5.0 and width 10.0: 0.14285714285714285 liters.
Computing amount of paint needed for Shape: Cylinder with radius 4.0 and height 10.0
Paint needed for Shape: Cylinder with radius 4.0 and height 10.0: 1.0053096491487339 liters.
BUILD SUCCESSFUL (total time: 0 seconds)

```

Figure 24 : Output

- d) The file PaintThings.java contains a program that computes the amount of paint needed to paint various shapes. A paint object has been instantiated. Add the following to complete the program: Instantiate the three shape objects: deck to be a 20 by 35 foot rectangle, bigBall to be a sphere of radius 15, and tank to be a cylinder of radius 10 and height 30. Make the appropriate method calls to assign the correct values to the three amount variables. Run the program and test it. You should see polymorphism in action as the amount method computes the amount of paint for various shapes.



The screenshot shows the NetBeans IDE interface with the following details:

- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, Shape..., Search (Ctrl+I).
- Project Explorer (Proj... X):** Shows the project structure with packages like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, and Source Packages containing files such as Cylinder.java, Main.java, Paint.java, and PaintThings.java.
- Code Editor (PaintThings.java):**

```

1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5
6 /**
7 *
8 * @author User
9 */
10 public class PaintThings {
11     public static void main(String[] args) {
12         // Step 1: Instantiate a Paint object with coverage of 350 sq. ft per gallon
13         Paint paint = new Paint(350);
14
15         // Step 2: Instantiate shape objects
16         Shape deck = new Rectangle(20, 35); // 20x35 foot rectangle
17         Shape bigBall = new Sphere(15); // Sphere with radius 15
18         Shape tank = new Cylinder(10, 30); // Cylinder with radius 10 and height 30
19
20         // Step 3: Compute the amount of paint needed for each shape
21         double deckPaint = paint.amount(deck);
22         double bigBallPaint = paint.amount(bigBall);
23         double tankPaint = paint.amount(tank);
24
25         // Step 4: Display results
26         System.out.println("\nAmount of paint required:");
27         System.out.printf("Deck (Rectangle): %.2f gallons\n", deckPaint);
28         System.out.printf("Big Ball (Sphere): %.2f gallons\n", bigBallPaint);
29         System.out.printf("Tank (Cylinder): %.2f gallons\n", tankPaint);
30     }
31 }

```
- Navigator (main - Navigator X):** Shows the members of the PaintThings class, including the main method.

Figure 25 : PaintThings Class

The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer (Proj... X):** Shows the project structure with packages like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages, Test Packages, Libraries, Test Libraries, and ShapePaint.
- Source Editor:** Displays the `PaintThings.java` file. The code implements a `Paint` class that calculates the amount of paint needed for different shapes (Rectangle, Sphere, Cylinder) based on their dimensions and a coverage rate of 350 sq. ft per gallon.
- Navigator X:** Shows the members of the `PaintThings` class, including the constructor `PaintThings()` and the main method `main(String[] args)`.
- Output - ShapeHierarchy (run):** Shows the terminal output of the application. It prints the amount of paint required for each shape and the total build time.

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license.
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5
6 /**
7 *
8 * @author User
9 */
10 public class PaintThings {
11     public static void main(String[] args) {
12         // Step 1: Instantiate a Paint object with coverage of 350 sq. ft per gallon
13         Paint paint = new Paint(350);
14
15         // Step 2: Instantiate shape objects
16         Shape deck = new Rectangle(20, 35); // 20x35 foot rectangle
17         Shape bigBall = new Sphere(15); // Sphere with radius 15
18         Shape tank = new Cylinder(10, 30); // Cylinder with radius 10 and height 30
19
20         // Step 3: Compute the amount of paint needed for each shape
21         double deckPaint = paint.amount(deck);
22         double bigBallPaint = paint.amount(bigBall);
23         double tankPaint = paint.amount(tank);
24
25         // Step 4: Display results
26         System.out.println("\nAmount of paint required:");
27         System.out.printf("Deck (Rectangle): %.2f gallons\n", deckPaint);
```

```
run:
Computing amount of paint needed for Shape: Rectangle with length 20.0 and width 35.0
Computing amount of paint needed for Shape: Sphere with radius 15.0
Computing amount of paint needed for Shape: Cylinder with radius 10.0 and height 30.0

Amount of paint required:
Deck (Rectangle): 2.00 gallons
Big Ball (Sphere): 8.08 gallons
Tank (Cylinder): 7.18 gallons
BUILD SUCCESSFUL (total time: 0 seconds)
```

Figure 26 : Output