



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

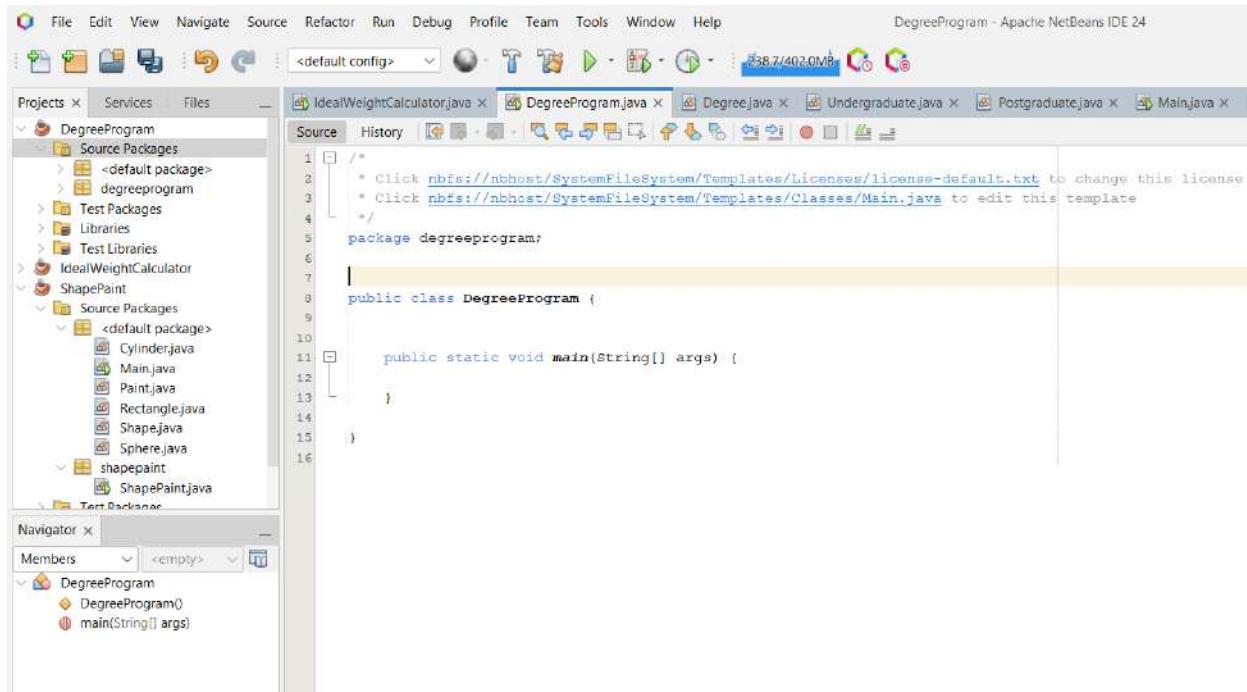
## **ECS2301-Software Engineering and Project**

**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.



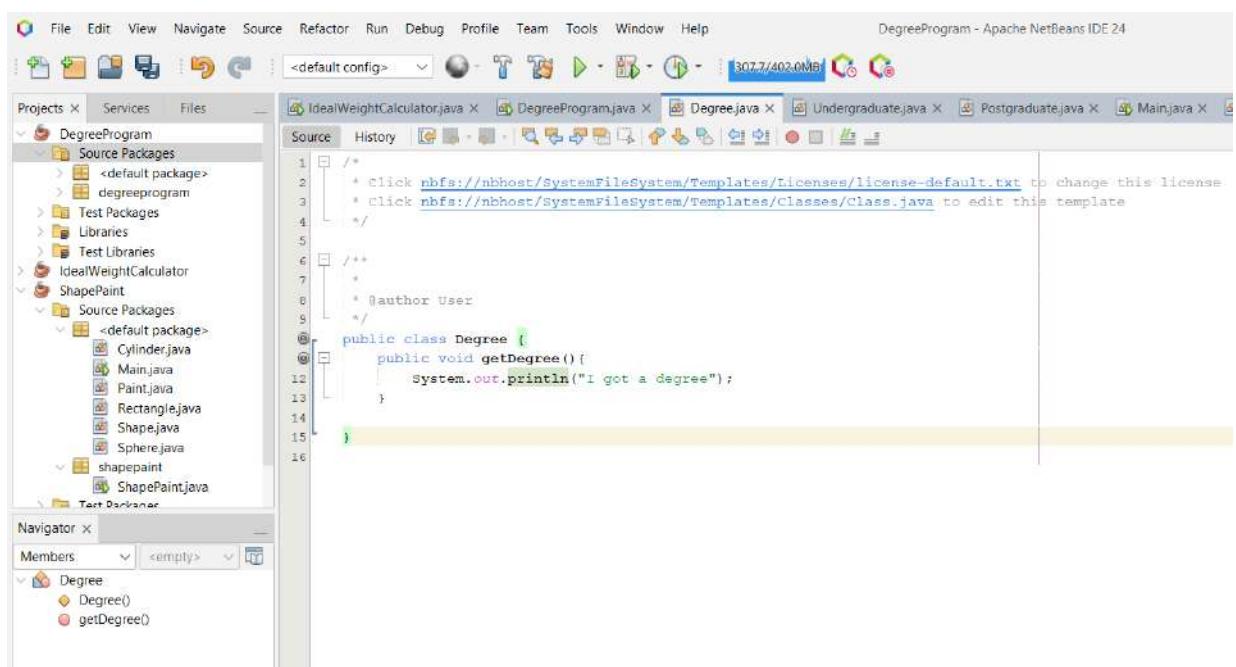
The screenshot shows the Apache NetBeans IDE 24 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, and Help. The title bar says "DegreeProgram - Apache NetBeans IDE 24". The Projects tab shows a tree view of the project structure under "DegreeProgram". The Source tab displays the code for "DegreeProgram.java". The Navigator tab shows the members of the "DegreeProgram" class, including the constructor "DegreeProgram()" and the main method "main(String[] args)".

```

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
8   public class DegreeProgram {
9
10    public static void main(String[] args) {
11        }
12    }
13
14
15 }
16

```

Figure 1: DegreeProgram 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 "DegreeProgram - Apache NetBeans IDE 24". The Projects tab shows a tree view of the project structure under "DegreeProgram". The Source tab displays the code for "Degree.java". The Navigator tab shows the members of the "Degree" class, including the constructor "Degree()" and the method "getDegree()".

```

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

```

Figure 2 : Degree 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, Help, and a status bar indicating 'DegreeProgram - Apache NetBeans IDE 24' with memory usage '202.8/402.0MB'. The Projects tab shows a hierarchy of packages: DegreeProgram (Source Packages: <default package>, degreeprogram, Test Packages, Libraries, Test Libraries), IdealWeightCalculator, ShapePaint (Source Packages: <default package> containing Cylinder.java, Main.java, Paint.java, Rectangle.java, Shape.java, Sphere.java, shapepaint, ShapePaint.java). The Navigator tab shows members of the Undergraduate class: Undergraduate (Undergraduate(), getDegree() + Degree). The Source tab displays the Undergraduate.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 Undergraduate extends Degree {
    public void getDegree() {
        System.out.println("I am an Undergraduate");
    }
}
```

Figure 3 : Undergraduate 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, Help, and a status bar indicating 'DegreeProgram - Apache NetBeans IDE 24' with memory usage '288.9/402.0MB'. The Projects tab shows a hierarchy of packages: DegreeProgram (Source Packages: <default package>, degreeprogram, Test Packages, Libraries, Test Libraries), IdealWeightCalculator, ShapePaint (Source Packages: <default package> containing Cylinder.java, Main.java, Paint.java, Rectangle.java, Shape.java, Sphere.java, shapepaint, ShapePaint.java, Test Packages). The Navigator tab shows members of the Postgraduate class: Postgraduate (Postgraduate(), getDegree() + Degree). The Source tab displays the Postgraduate.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 Postgraduate extends Degree {
    public void getDegree() {
        System.out.println("I am a Postgraduate");
    }
}
```

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 says "DegreeProgram - Apache NetBeans IDE 24". The Projects tab is selected, showing a project named "DegreeProgram" with several source packages: "Source Packages", "Test Packages", "Libraries", "Test Libraries", "IdealWeightCalculator", and "ShapePaint". The "ShapePaint" package contains sub-packages like "Source Packages", "Test Packages", "Libraries", and "Test Libraries". The "Source Packages" folder under "ShapePaint" contains classes: Cylinder.java, Main.java, Paint.java, Rectangle.java, Shape.java, Sphere.java, ShapePaint.java, and TextPainter.java. The main editor window displays the "Main.java" file:

```
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         Degree degree = new Degree();
13         degree.getDegree(); // Outputs: I got a degree
14
15         Undergraduate ug = new Undergraduate();
16         ug.getDegree(); // Outputs: I am an Undergraduate
17
18         Postgraduate pg = new Postgraduate();
19         pg.getDegree(); // Outputs: I am a Postgraduate
20     }
21
22 }
```

The Navigator pane shows the members of the Main class: Main(), Main(), and main(String[] args). The status bar at the bottom indicates "173.2/4020MB".

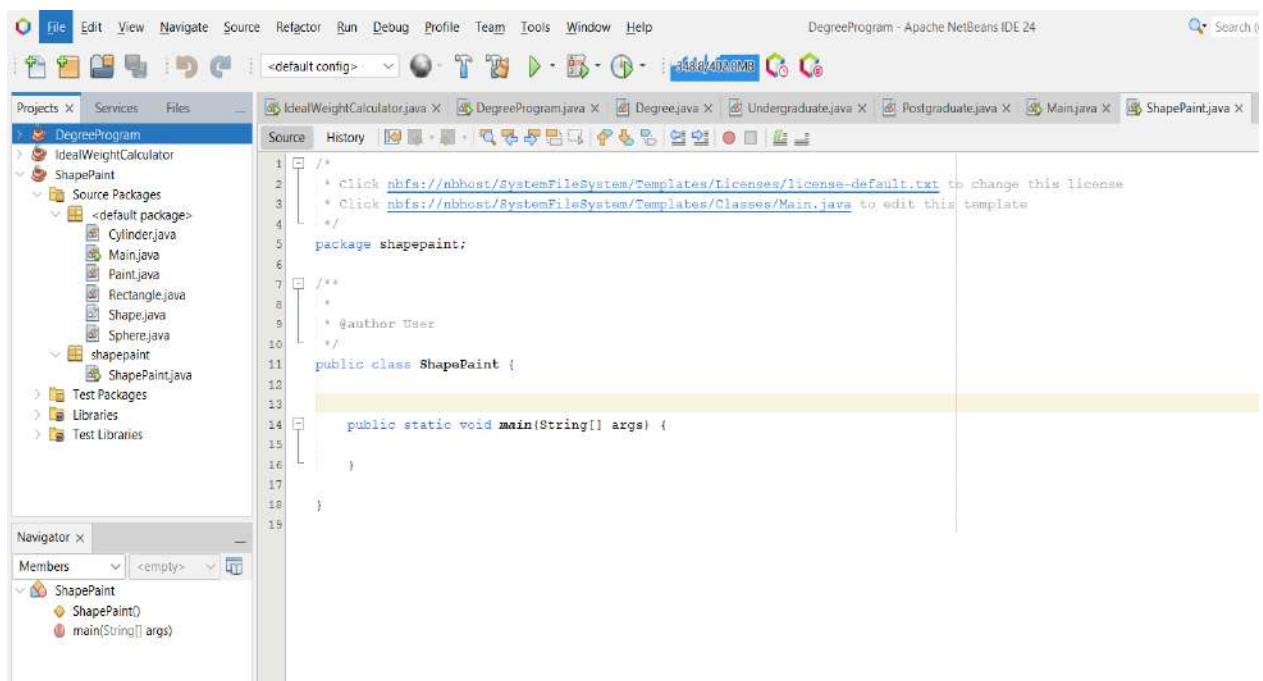
Figure 5 : Main Class

This screenshot shows the same Apache NetBeans IDE 24 interface as Figure 5, but it also includes the Output window at the bottom. The Output window is titled "Output - DegreeProgram (run)" and displays the following text:

```
run:
I got a degree
I am an Undergraduate
I am a Postgraduate
BUILD SUCCESSFUL (total time: 0 seconds)
```

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



```

 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

```

Figure 7 : ShapePaint Class

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

- Projects View:** Shows the project structure with packages like DegreeProgram, IdealWeightCalculator, and ShapePaint.
- Source Editor:** Displays the `Shape.java` file content:

```
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 }
```
- Navigator View:** Shows the members of the `Shape` class, including `area(): double`, `toString(): String`, and `shapeName: String`.

Figure 8 : Shape Class

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

- Projects View:** Shows the project structure with packages like DegreeProgram, IdealWeightCalculator, and ShapePaint.
- Source Editor:** Displays the `Rectangle.java` file content:

```
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 }
```
- Navigator View:** Shows the members of the `Rectangle` class, including `area(): double`, `length: double`, and `width: double`.

Figure 9 : Rectangle Class

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

- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer (Projects X):** DegreeProgram, IdealWeightCalculator, ShapePaint.
- Source Editor:** Java file `Sphere.java` open. The code defines a `Sphere` class extending `Shape`. It has a private double `radius` and an overridden `area()` method that returns  $4 * \pi * r^2$ .
- Navigator:** Shows members of the `Sphere` class: `Sphere`, `Sphere(double radius)`, `area(): double`, `Shape`, and `radius: double`.

Figure 10 : Sphere Class

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

- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Project Explorer (Projects X):** DegreeProgram, IdealWeightCalculator, ShapePaint.
- Source Editor:** Java file `Cylinder.java` open. The code defines a `Cylinder` class extending `Shape`. It has private double `radius` and `height`. The constructor initializes these variables. The `area()` method calculates the surface area of a cylinder as  $2\pi r(r + h)$ .
- Navigator:** Shows members of the `Cylinder` class: `Cylinder`, `Cylinder(double radius, double height)`, `area(): double`, `height: double`, and `radius: double`.

Figure 11 : Cylinder Class

The screenshot shows the Apache NetBeans IDE 24 interface. The left pane displays the project structure under 'Projects X' for 'ShapePaint'. The right pane shows the source code for the 'Paint.java' file. The code defines a class 'Paint' with a private attribute 'coverage' and two methods: 'Paint(double coverage)' and 'amountNeeded(Shape shape)'. The 'amountNeeded' method returns the area of the shape divided by the coverage.

```

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 }
21
22
23
24
25
26
27

```

Figure 12 : Paint Class

The screenshot shows the Apache NetBeans IDE 24 interface. The left pane displays the project structure under 'Projects X' for 'ShapePaint'. The right pane shows the source code for the 'Main.java' file. The code contains a main method that creates instances of 'Shape' objects (rectangle, sphere, cylinder) and prints the amount of paint needed for each.

```

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
26
27

```

Figure 13 : Main Class

The screenshot shows the Apache NetBeans IDE 24 interface. The Projects panel on the left lists several projects: DegreeProgram, IdealWeightCalculator, ShapePaint, Test Packages, Libraries, and Test Libraries. The ShapePaint project is expanded, showing Source Packages, <default package>, Main.java, Cylinder.java, Paint.java, Rectangle.java, Shape.java, and Sphere.java. The Navigator panel shows the members of Main.java, including Main0 and main(String[] args). The Source editor 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
 */
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);

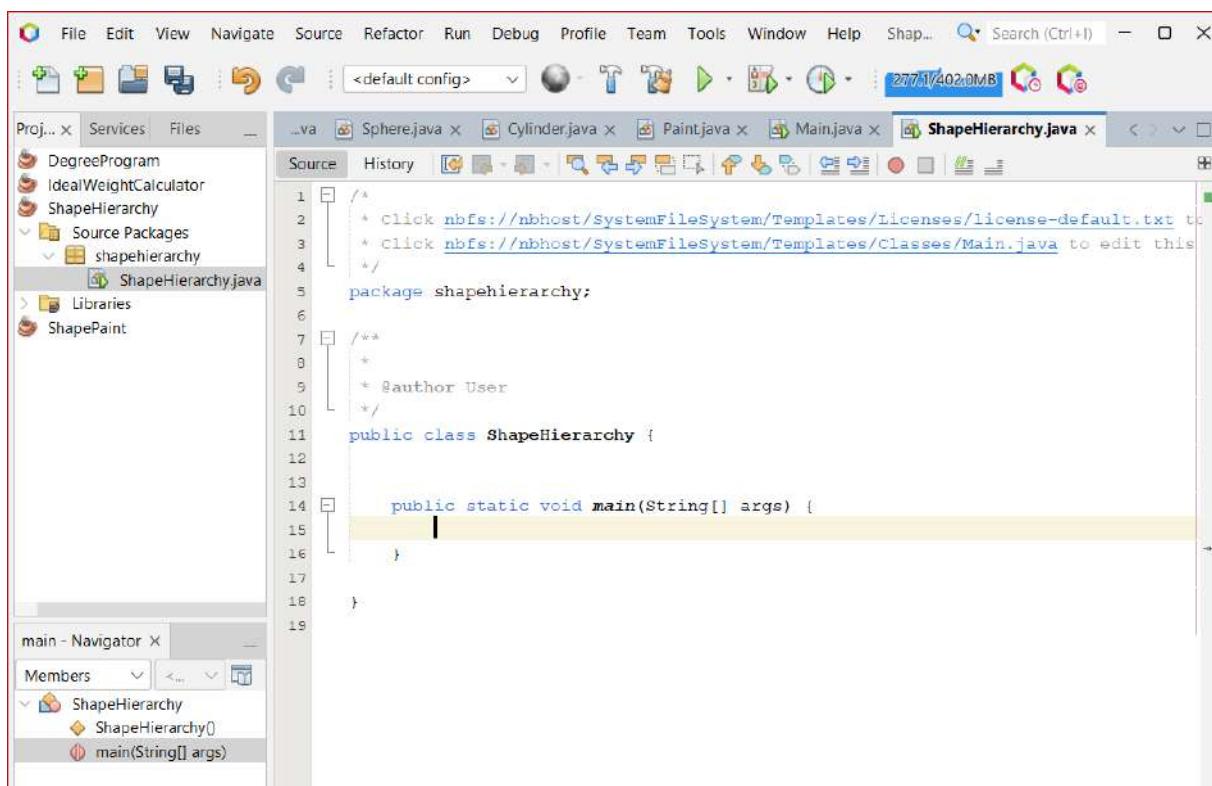
        // Calculate and display amount of paint needed
        System.out.println(rectangle + " needs " + paint.amountNeeded(rectangle) + " liters of paint.");
        System.out.println(sphere + " needs " + paint.amountNeeded(sphere) + " liters of paint.");
        System.out.println(cylinder + " needs " +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

The Output - ShapePaint (run) window at the bottom shows the program's output:

```
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. The ShapeHierarchy.java file is selected.
- Code Editor (Center):** Displays the `ShapeHierarchy.java` source code. The code defines a package named `shapehierarchy` and a public class `ShapeHierarchy` with a main method.
- Navigator (Bottom Left):** Shows the members of the `ShapeHierarchy` class, including the constructor `ShapeHierarchy()` and the `main` method.
- Toolbar (Top):** Includes standard Java development tools like class navigation, code completion, and run/debug buttons.
- Status Bar (Bottom):** Shows memory usage (2771/4020MB) and other system information.

```

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 }

```

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, Libraries, and ShapePaint.
- Code Editor:** Displays the `Shape.java` file content:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
 * click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
 */
/*
 *
 * @author User
 */
public abstract class Shape {
    protected String shapeName;

    public Shape(String shapeName) {
        this.shapeName = shapeName;
    }

    // Abstract method to compute area
    public abstract double area();

    @Override
    public String toString() {
        return "Shape: " + shapeName;
    }
}
```
- Navigator:** Shows members of the `Shape` class, including `shapeName`, `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, Test Packages, Libraries, Test Libraries, and ShapePaint.
- Code Editor:** Displays the `Sphere.java` file content:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
 * click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
 */
/*
 *
 * @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;
    }

    @Override
    public String toString() {
        return super.toString() + " with radius " + radius;
    }
}
```
- Navigator:** Shows members of the `Sphere` class, including `radius`, `area()`, and `toString()`.

Figure 17 : Sphere Class

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

- Project Explorer (Proj... X):** Shows files like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages, Test Packages, Libraries, Test Libraries, and ShapePaint.
- Code Editor (Source tab):** Displays the `Rectangle.java` file content. The code defines a `Rectangle` class that extends `Shape`. It has private fields `length` and `width`, a constructor, and overrides for `area()` and `toString()`.
- Navigator (toString - Navigator X):** Shows the members of the `Rectangle` class, including its constructor, `area()`, `toString()`, and its two fields `length` and `width`.

```

1 /**
 * 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
 */
2
3
4
5
6 /**
7 *
8 * @author User
9 */
10
11 public class Rectangle extends Shape {
12     private double length;
13     private double width;
14
15     public Rectangle(double length, double width) {
16         super("Rectangle");
17         this.length = length;
18         this.width = width;
19     }
20
21     @Override
22     public double area() {
23         return length * width;
24     }
25
26     @Override
27     public String toString() {
28         return super.toString() + " with length " + length + " and width " + width;
29     }
30 }

```

Figure 18 : Rectangle Class

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

- Project Explorer (Proj... X):** Shows files like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages, Test Packages, Libraries, Test Libraries, and ShapePaint.
- Code Editor (Source tab):** Displays the `Cylinder.java` file content. The code defines a `Cylinder` class that extends `Shape`. It has private fields `radius` and `height`, a constructor, and overrides for `area()` and `toString()`.
- Navigator (toString - Navigator X):** Shows the members of the `Cylinder` class, including its constructor, `area()`, `toString()`, and its two fields `radius` and `height`.

```

1 /**
 * 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
 */
2
3
4
5
6 /**
7 *
8 * @author User
9 */
10
11 public class Cylinder extends Shape{
12     private double radius;
13     private double height;
14
15     public Cylinder(double radius, double height) {
16         super("Cylinder");
17         this.radius = radius;
18         this.height = height;
19     }
20
21     @Override
22     public double area() {
23         // Surface area of a cylinder = 2πr(r + h)
24         return 2 * Math.PI * radius * (radius + height);
25     }
26
27     @Override
28     public String toString() {
29         return super.toString() + " with radius " + radius + " and height " + height;
30     }
31 }

```

Figure 19 : Cylinder Class

The screenshot shows the NetBeans IDE interface with the Main.java file open in the editor. The code defines a Main class with a main method that creates instances of Shape, Rectangle, and Cylinder and prints their areas.

```
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 }

```

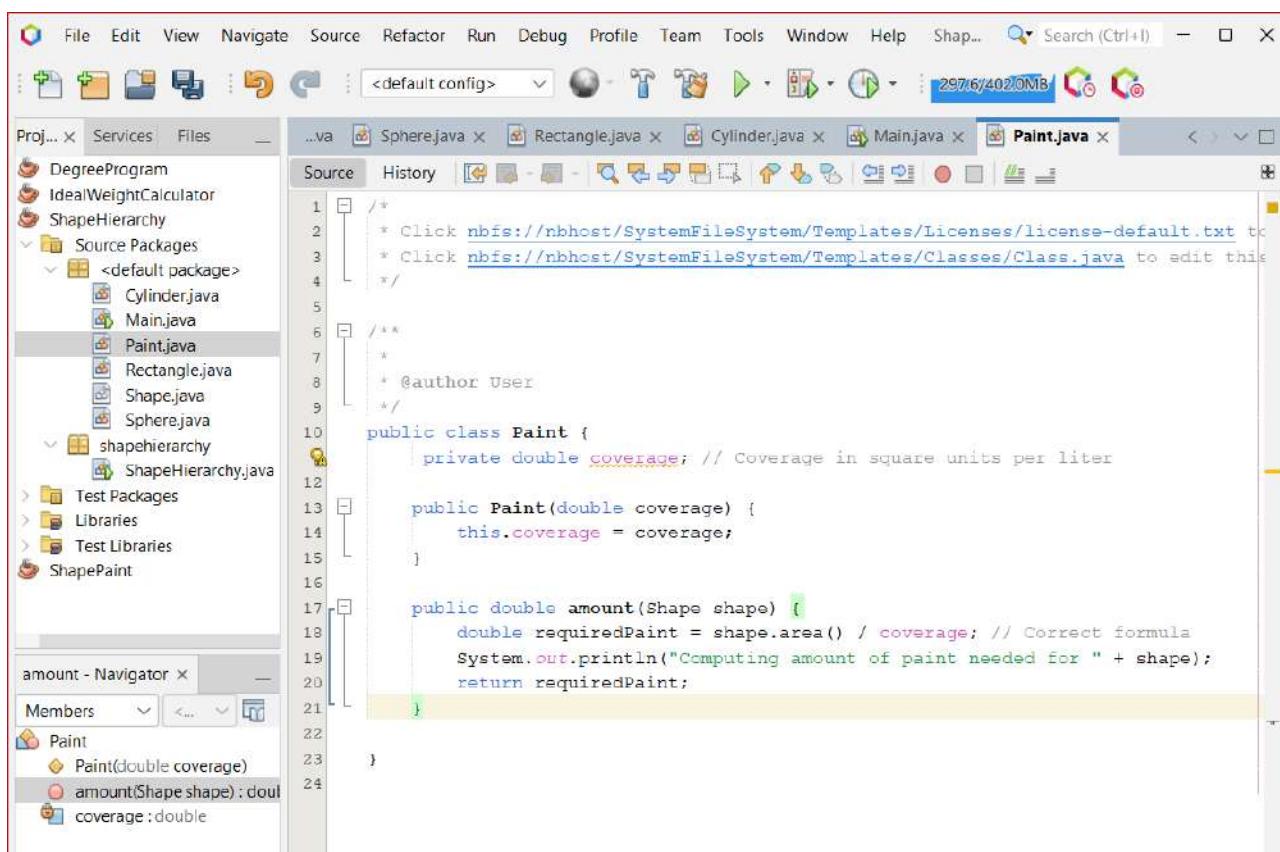
Figure 20 : Main Class

The screenshot shows the NetBeans IDE interface with the Main.java file open in the editor. The code is identical to Figure 20. Below the editor, the Output window displays the program's 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.)



The screenshot shows the NetBeans IDE interface with the Paint.java file open in the editor. The code defines a class Paint with a constructor and an amount method. The amount method currently returns a constant value of 1.0, which is incorrect. The correct formula is  $\text{area} / \text{coverage}$ .

```

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   * @author User
8   */
9
10 public class Paint {
11     private double coverage; // Coverage in square units per liter
12
13     public Paint(double coverage) {
14         this.coverage = coverage;
15     }
16
17     public double amount(Shape shape) {
18         double requiredPaint = shape.area() / coverage; // Correct formula
19         System.out.println("Computing amount of paint needed for " + shape);
20         return requiredPaint;
21     }
22
23 }
24

```

Figure 22 : Paint Class

The screenshot shows the Apache NetBeans IDE interface. The Project Explorer on the left lists several Java files: Cylinder.java, Paint.java, Main.java, ShapeHierarchy.java, and others. The Source tab in the center editor window displays the Main.java code. The code defines a Main class with a main method that creates instances of Shape (Sphere, Rectangle, Cylinder) and prints their areas. It then calculates and prints the amount of paint needed for each shape based on a coverage of 350 square units per liter.

```

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

This screenshot shows the Apache NetBeans IDE with the Main.java code in the editor and the output of the run command in the Output window below. The output window displays the areas of the shapes and the amounts of paint required for each.

```

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: 1.758251886010284 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, Search (Ctrl+I).
- Project Explorer (left):** Shows the project structure with packages like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages, and Test Packages. The file PaintThings.java is selected.
- Code Editor (center):** Displays the Java code for PaintThings.java. The code defines a main method that creates a Paint object (coverage 350 sq. ft per gallon), instantiates four shape objects (Rectangle, Sphere, and two Cylinder), computes the amount of paint needed for each, and prints the results.

```
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   * @author User
8   */
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("Amount 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
32 }
```

- Navigator (bottom left):** 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 packages like DegreeProgram, IdealWeightCalculator, ShapeHierarchy, Source Packages (<default package> containing Cylinder.java, Main.java, Paint.java, PaintThings.java), Test Packages, Libraries, Test Libraries, and ShapePaint.
- Source Editor:** Displays the Java code for `PaintThings.java`. The code implements a `Paint` object to calculate paint requirements for different shapes (Rectangle, Sphere, Cylinder) based on their dimensions and a coverage rate of 350 sq. ft per gallon.
- Navigator X:** Shows members of the `PaintThings` class, including the constructor `PaintThings()` and the main method `main(String[] args)`.
- Output Window:** Shows the run output for `ShapeHierarchy`, displaying the amount of paint required for each shape: Deck (Rectangle): 2.00 gallons, Big Ball (Sphere): 8.08 gallons, and Tank (Cylinder): 7.18 gallons. It also shows a successful build message.

```
/*
 * 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 PaintThings {
    public static void main(String[] args) {
        // Step 1: Instantiate a Paint object with coverage of 350 sq. ft per gallon
        Paint paint = new Paint(350);

        // Step 2: Instantiate shape objects
        Shape deck = new Rectangle(20, 35); // 20x35 foot rectangle
        Shape bigBall = new Sphere(15); // Sphere with radius 15
        Shape tank = new Cylinder(10, 30); // Cylinder with radius 10 and height 30

        // Step 3: Compute the amount of paint needed for each shape
        double deckPaint = paint.amount(deck);
        double bigBallPaint = paint.amount(bigBall);
        double tankPaint = paint.amount(tank);

        // Step 4: Display results
        System.out.println("\nAmount of paint required:");
        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