**LAB MANUAL**



**ROLLNO:AV.SC.U4CSE24322**

**NAME: S .SAI VENKAT**

**SECTION: CSE-B**

**WEEK-1:**

**Aim:** How to install jdk and first program on

printing student details*.*

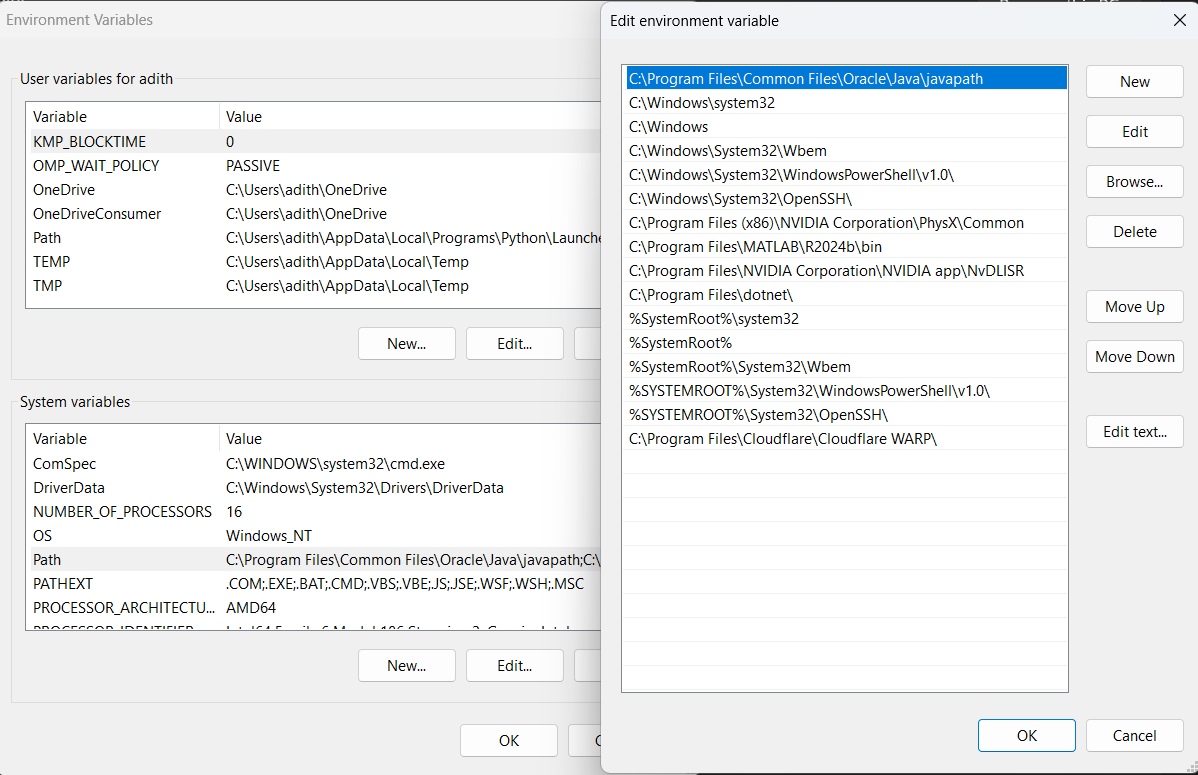
**Step-1:** Download JDK-21 from oracle website

**

**Step-2:**Install the JDK-21 with accepting terms and

conditions according to the respective windows.

**Step-3**:Setting up environmental variables.



\*Windows c -> C-drive -> program files ->Java -

>JDK-21->select bin

\*Select and open environmental variable in search

bar-> either select system variables or user

variables-> select path-> click edit->New-> paste

the bin-> finish the setup(apply the changes).

~for verifying the installed version

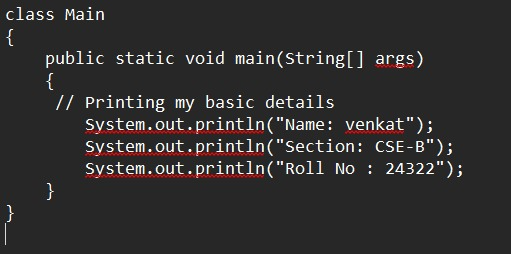
Open cmd-> type java --version

~command propt

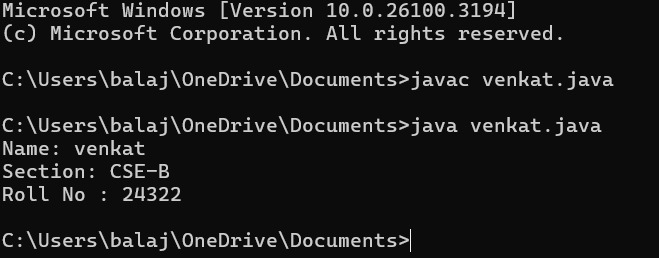
Javac filename.java ->compiling.

Java filename.java ->displaying

**PROGRAM-1(Rectified):**

******

**Output:**

***s***

**WEEK-2:**

**PROGRAM-1:**

**Aim:**Write a java program for SI

**

**Output:**

******

**ERROR TABLE*:***

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1.Giving space between next and Double.  2.Not giving parenthesis after closing the input. | 1.Should not give space between next and Double.  2.We must put parenthesis after closing the input. |

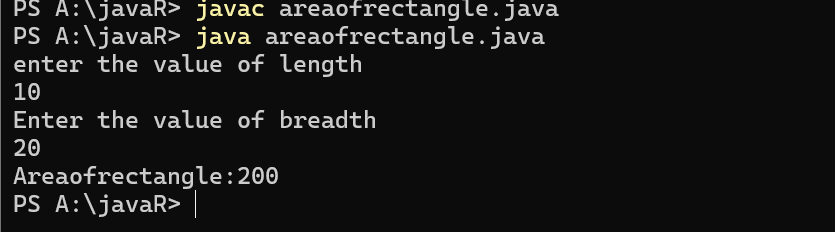
**PROGRAM-2:**

**Aim:**Write a program in java for area of

rectangle.

**

**Output:**

******

**ERROR TABLE:**

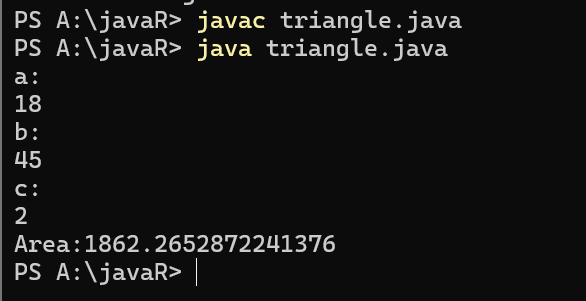
|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1.While using for iteration, not giving the conditions correctly.  2.Declaring the data type as double instead of int. | 1.We should give iterative statements correctly.  2.We should give the data type as int for integers. |

**PROGRAM-3:**

**Aim:**Write a program in java for area of triangle using heron’s formula.



**Output:**

******

ERROR TABLE:

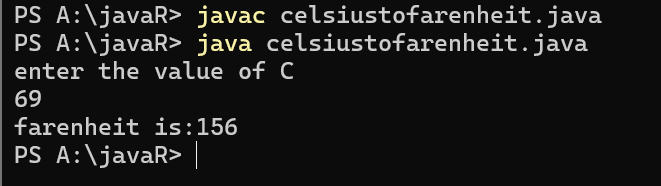
|  |  |
| --- | --- |
| **Code Error** | Code rectification |
| 1.While printing the variable not giving + sign.  2.Not closing the scanner. | 1.We should give correct indentation.  2.Closing the scanner is must. |

**PROGRAM-4(a):**

**Aim:**Write a program in java for converting temperature from celsius to fahrenite.



OUTPUT:

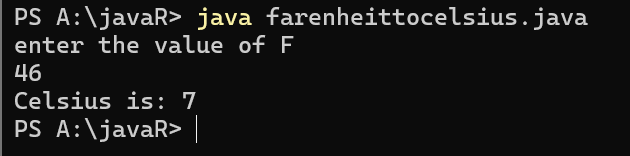


**PROGRAM-4(b):**

**Aim:**Write a program in java for converting temperature from fahrenite to celsius.

******

**Output:**

******

**ERROR TABLE:**

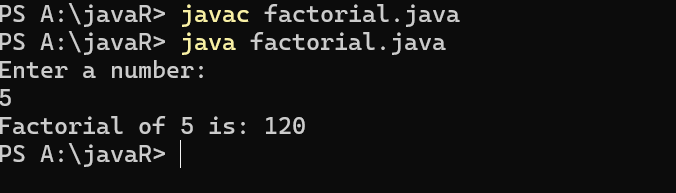
|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1.While printing the variable not giving + sign.  2.Not closing the scanner. | 1.We should give correct indentation.  2.Closing the scanner is must. |

**PROGRAM-5:**

**Aim:**Write a program in java for factorial of a number.

******

OUTPUT:



ERROR TABLE:

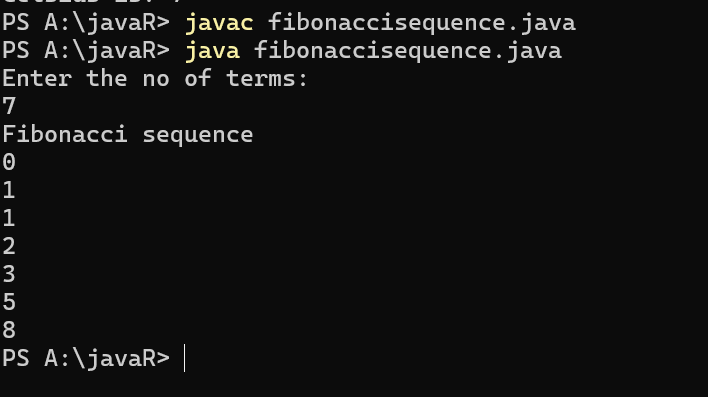
|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1.While using for iteration, not giving the conditions correctly.  2.Declaring the data type as double instead of int. | 1.We should give iterative statements correctly.  2.We should give the data type as int for integers. |

**PROGRAM-6:**

**Aim:**Write a program in java for fibonacci series.



OUTPUT:



ERROR TABLE:

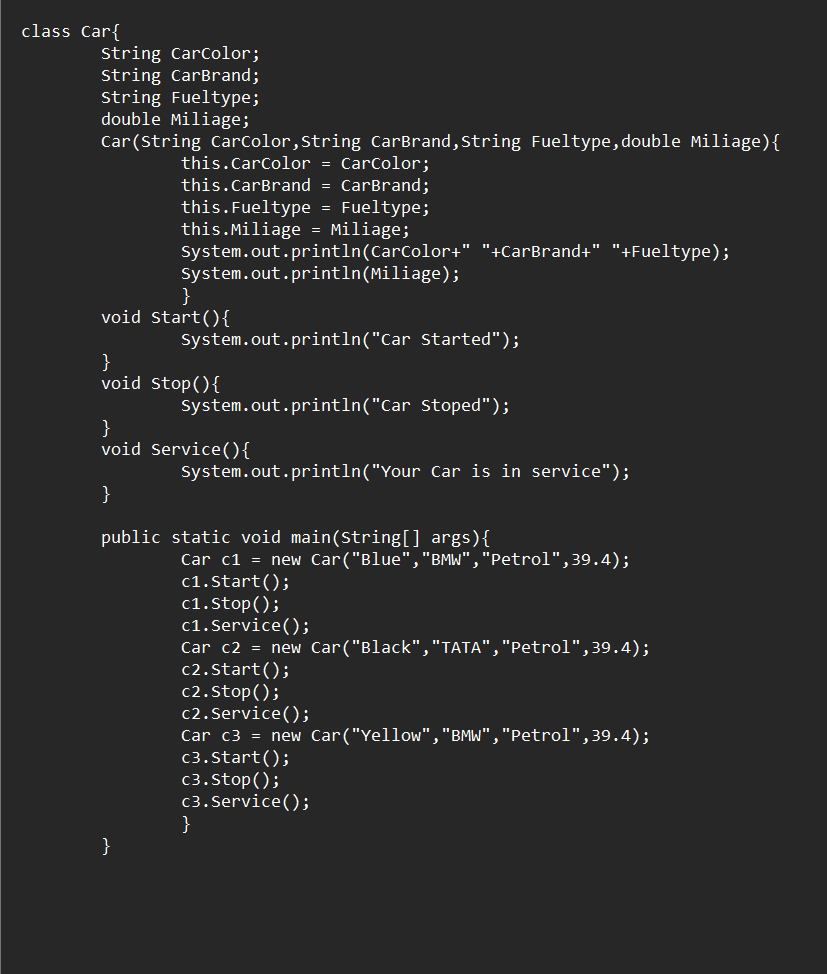
|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1.Giving space between next and Double.  2.Not giving parenthesis after closing the input. | 1.Should not give space between next and Double.  2.We must put parenthesis after closing the input. |

**Week 3**

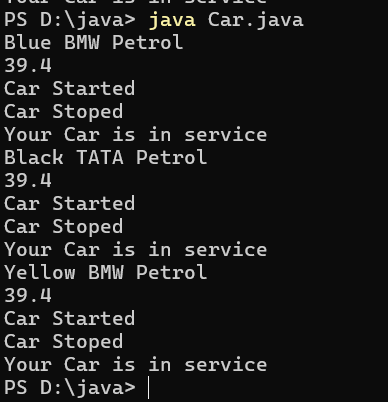
**Program 1:**

**Aim :** Write a java program with the following instructions

1. Create a class with name “Car”
2. Create 4 attributes named CarColor,CarBrand,Fueltype,Miliage
3. Create 3 methods named Start,Stop,Service
4. Create 3 objects named c1,c2,c3
5. Create a constructor with parameters CarColor,CarBrand,Fueltype,Miliage



**Output :**

****

ERROR TABLE:

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1.I have kept ‘,’ between variables in print statement | 1.Insted of ‘,’ we should use ‘+’ between  variables in print statement |

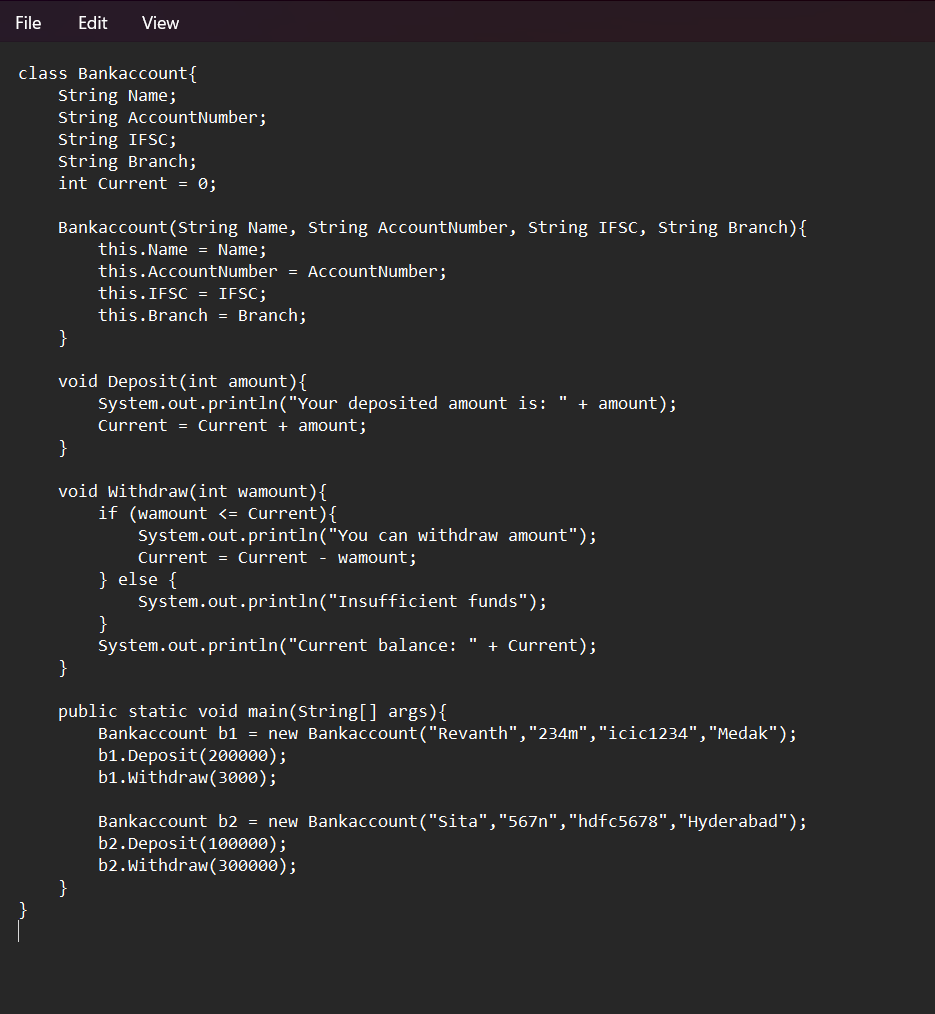
**Program 2:**

**Aim:** Create a class named bank account with methods Deposit,withdraw,were the deposit method should accepts a parameter and when this method is called the deposited amount should added to current balance, in addition to that when a withdraw method is called it has to verify where current balance, if current balance is less then withdraw amount , then “There are insufficient funds” message should display.

🡪Use the constructor to display the details of the customer,(Name,AccountNumber,IFSE,Branch)

🡪Also Create two customer objects

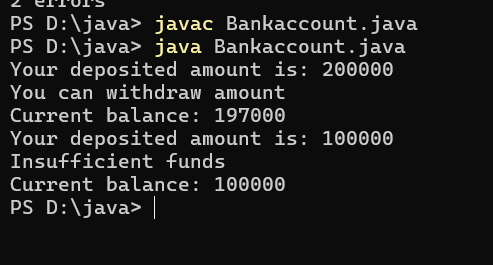
**Program:**

****

ERROR TABLE:

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. The condition checking in the withdrawal amount should be <= | 1. Change the condition to correct form |

**Output:**



**WEEK-4:**

**PROGRAM-1:**

**Aim:** Write a java program with class named book .The class should contain various attributes such as Title ,Author and Year of Publication .It should also contain a constructor with parameter which initializes Title ,Author and Year of publication .Create a method which displays the details of the book .Display the details of two books.

**CODE:**

class book{

public String title;

public String author;

public String year\_of\_publication;

public void book(){

this.title=title;

this.author=author;

this.year\_of\_publication=year\_of\_publication;

}

public static void main(String[] args){

book book1=new book();

book book2=new book();

book1.book();

book1.title="Sherlock Holmes ";

book1.author="Arthur Conan Doyle";

book1.year\_of\_publication="1887";

book2.book();

book2.title="Harry Potter";

book2.author="J.K. Rowling";

book2.year\_of\_publication="1997";

System.out.println("Book-1");

System.out.println("Title :" +book1.title);

System.out.println("Author :" +book1.author);

System.out.println("Year of publication :" +book1.year\_of\_publication);

System.out.println("Book-2");

System.out.println("Title :" +book2.title);

System.out.println("Author :" +book2.author);

System.out.println("Year of publication :" +book2.year\_of\_publication);

}

}

**OUTPUT:**

****

|  |
| --- |
| **Book** |
| * Title: String * Author: String * Year of publication: int |
| + Book(title: String,  Author: String;  Year of publication: int  + displayDetails( ): void |

**Class Diagram:**

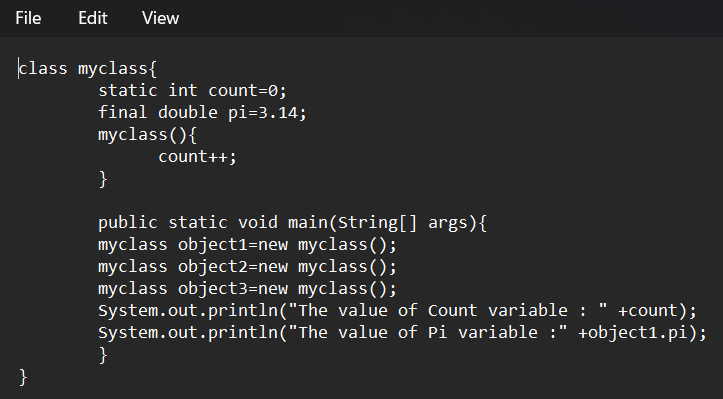
IMPORTANT POINTS:

1. While defining two classes for a code, we must be sure that we save both the classes in separate files.
2. While defining a method we should also define a function to call that method.

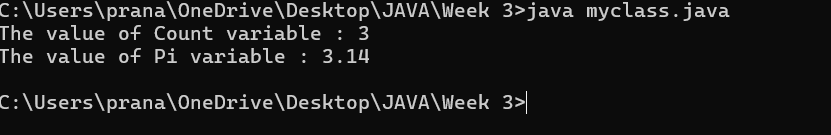
**Error Table:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Error** | **Rectification** |
| **1.** | **Missing “;” after calling method.** | **Added “;”** |

**PROGRAM-2:**

**Aim:** Create a java program with class named myclass with a static variable count of int type initialized to 0 and a constant variable “Pi” of type double initialized to 3.14 has attributes of that class .Now define a constructor for “myclass” that incerements the count variable each time an object of myclass is created finally print final values count and Pi variables .Create three objects

**OUTPUT:**

****

**Class Diagram:**

|  |
| --- |
| Myclass |
| - Count: int  - Pi: double |
| + myclass( )  + main(args: String[]): void |

IMPORTANT POINTS:

1. We must declare the initial value of the variable before declaring the final one.
2. Here the main objective is to increase the count according to the number of objects we make, i.e the count increases when the no.of objects are increasing.

Error Table:

|  |  |  |
| --- | --- | --- |
| S.No | Error | Rectification |

|  |  |  |
| --- | --- | --- |
| 1. | Not typing “}“ at the end of the code. | Added “}”. |

**Class diagram:**

|  |
| --- |
| Myclass |
| Count : int  PIE : double |
| +Myclass() |

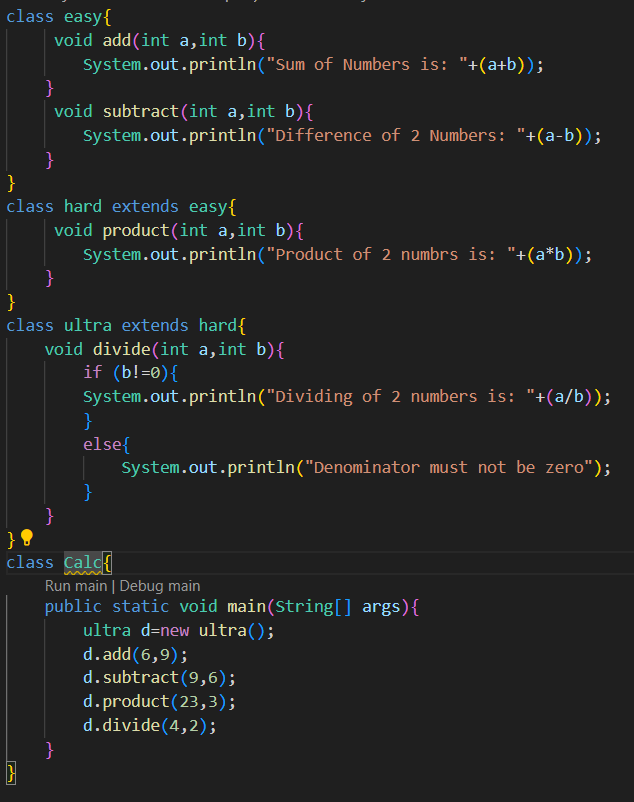
**WEEK-5  
  
1) Create a calculator using the operations including addition, subtraction, multiplication and division using multilevel in heritance and display the desired output.**

**- Write your code in VS CODE and execute**

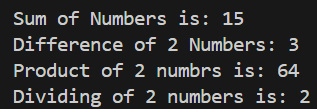
**- Important Points:**

* + - * **Understand the calling of a Constructor**
      * **Giving class name correctly**
      * **Give the parameters Correctly**

**CODE:**

****

**OUTPUT:**

 **Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| 1 | Syntax/ Compilation Error | Absence of Semicolon |
| 2 | Closing Brackets | Need to Close the brackets |
| 3 | Class Name Error | Give the class name correctly |
| 4 | Constructor Calling | Call the constructor correctly |

**2) Vehicle rental company wants to develop a system that maintains information about different types of vehicles available for rent. The company rents out cars and bikes and they need a program to store details about each vehicle such as brand and speed.**

**i. Cars should have an additional property: number of doors, Seating capacity.**

**ii. Bikes should have a property indicating whether they have gears or not.**

**iii. The system should also include a function to display details about each vehicle and indicate when a vehicle is starting.**

**iv. Each class should have a constructor.**

**Questions:**

**1. Which OOP concept is used in the above program? Explain why it is useful in this scenario.**

**2. If the company decides to add a new type of vehicle ‘Truck’, how would you modify the program?**

**a. Truck should include and additional property capacity (in tons).**

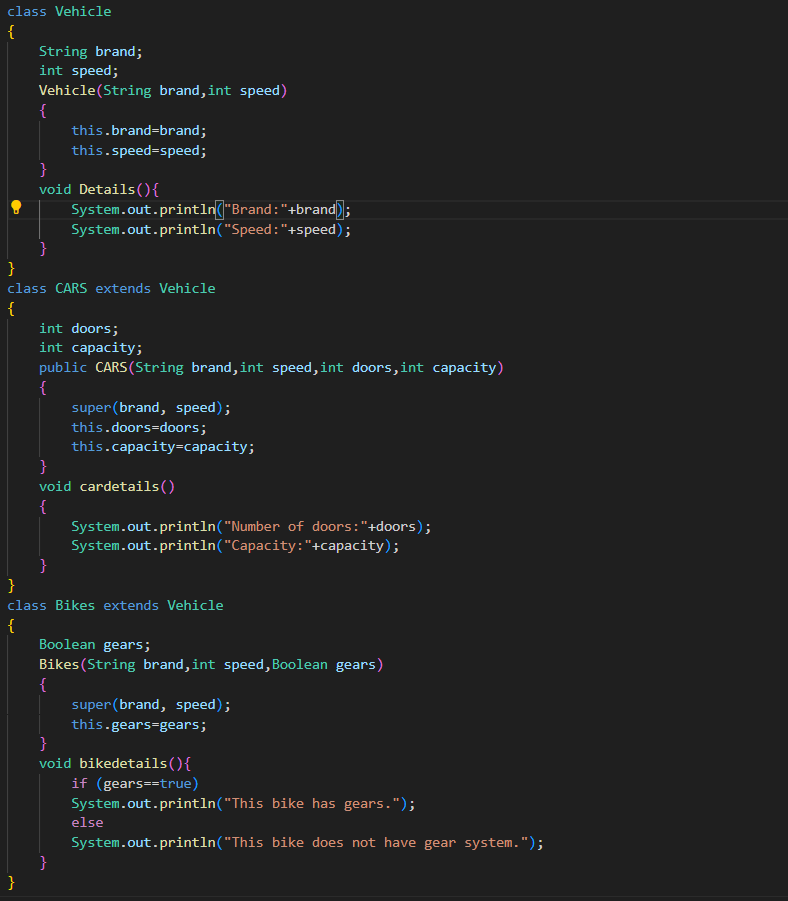
**b. Create a showTruck() method to display the truck’s capacity.**

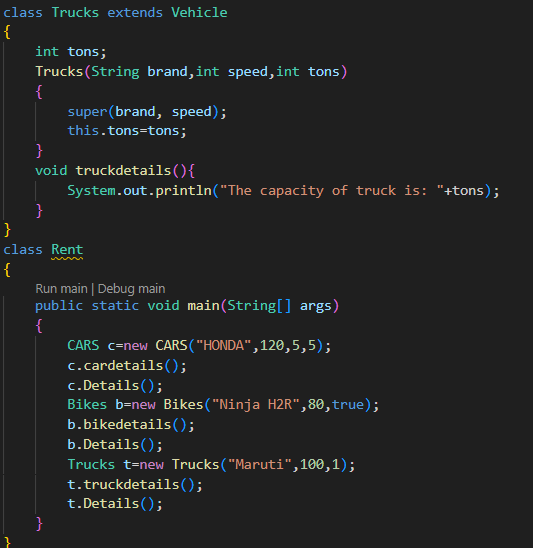
**c. Write a constructor for truck that initializes all properties.**

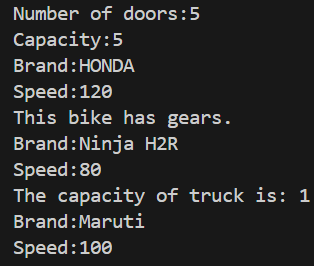
**3. Implement the truck class and update the main method to create a Truck object and also create an object for car and bike subclasses. Finally display the details.  
  
  
  
- Write your code in VS CODE and execute**

**- Important Points:**

* + - * **Understand the calling of a Constructor**
      * **Giving class name correctly**
      * **Give the parameters Correctly**

**CODE:  
**



**OUTPUT:**

**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| 1 | Syntax/ Compilation Error | Absence of Semicolon |
| 2 | Closing Brackets | Need to Close the brackets |
| 3 | Class Name Error | Give the class name correctly |
| 4 | Constructor Calling | Call the constructor correctly |

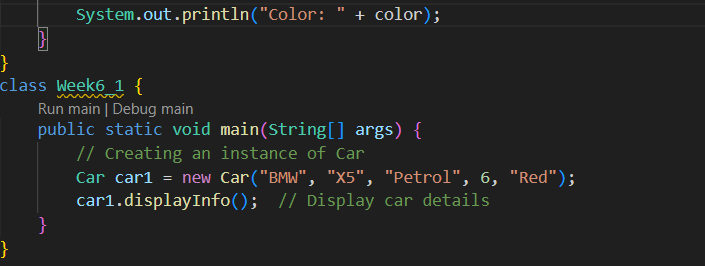
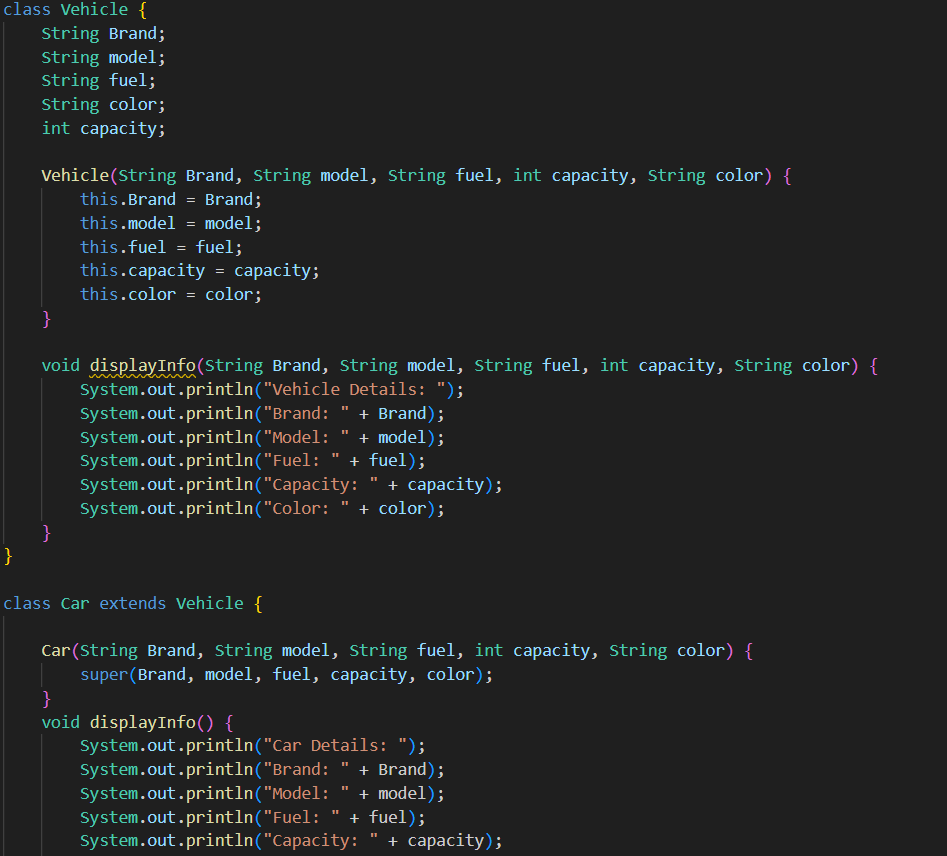
**WEEK-6**

1) **Write a Java program to create a vehicle class with a method displayInfo(). Override this method in the car subclass to provide specific information about a car, model, fuel type, and color using the constructor**

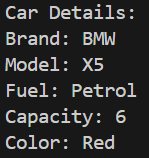
- Write your code in VS CODE and execute

**- Important Points:**

* + - * Understand the calling of a Constructor
      * Giving class name correctly
      * Give the parameters Correctly

**CODE:**  


**OUTPUT:**



**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| 1 | Syntax/ Compilation Error | Absence of Semicolon |
| 2 | Closing Brackets | Need to Close the brackets |
| 3 | Class Name Error | Give the class name correctly |
| 4 | Constructor Calling | Call the constructor correctly |

**2) Create a Java program for the scenario.**

**A college is developing an automated admission system that verifies student eligibility for undergraduate (UG) and postgraduate(PG) programs. Each program has different eligibility criteria based on the student's percentage in their previous qualification.**

**i) UG admissions require a minimum of 60%**

**ii) PG admissions require a minimum of 70%**

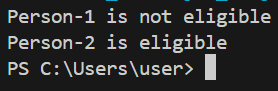
**- Write your code in VS CODE and execute**

**- Important Points:**

* + - * **Understand the calling of a Constructor**
      * **Giving class name correctly**
      * **Give the parameters Correctly**

**CODE:**

**  
  
  
  
  
  
OUTPUT:**

****

**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| 1 | Syntax/ Compilation Error | Absence of Semicolon |
| 2 | Closing Brackets | Need to Close the brackets |
| 3 | Class Name Error | Give the class name correctly |
| 4 | Constructor Calling | Call the constructor correctly |

**3) Write a Java Program to create a Calculator class with overloaded methods to perform addition: Take the integer values a and b from the user.**

**i) Add two integers**

**ii) Add two doubles**

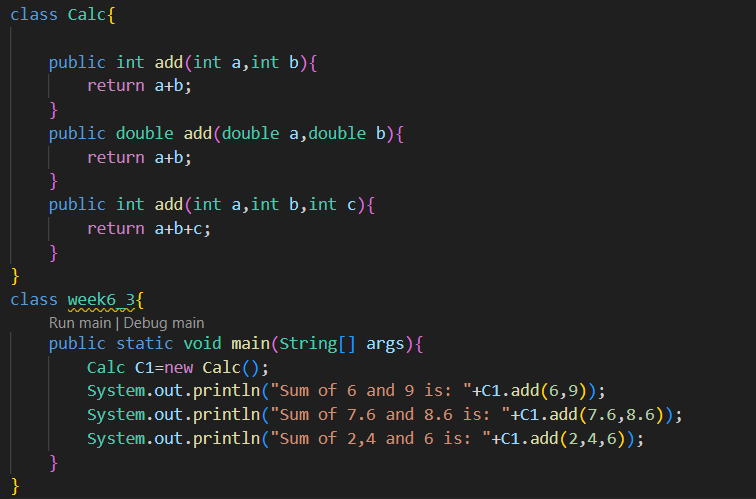
**iii) Add three integers**

**- Write your code in VS CODE and execute**

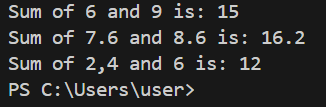
**- Important Points:**

* + - * **Understand the calling of a Constructor**
      * **Giving class name correctly**
      * **Give the parameters Correctly**

**CODE:**

****

**OUTPUT:**

****

**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| 1 | Syntax/ Compilation Error | Absence of Semicolon |
| 2 | Closing Brackets | Need to Close the brackets |
| 3 | Class Name Error | Give the class name correctly |
| 4 | Constructor Calling | Call the constructor correctly |

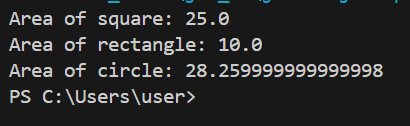
**4) Write a Java Program to create a shape class with a method calculateArea() that is overloaded for different shapes(e.g., Square, Rectangle ). Then create a subclass Circle that overrides the calculateArea() method for a circle.**

- Write your code in VS CODE and execute

**- Important Points:**

* + - * Understand the calling of a Constructor
      * Giving class name correctly
      * Give the parameters Correctly

**CODE:  
  
  
  
  
  
  
  
  
OUTPUT:**

****

**Errors:**

|  |  |  |
| --- | --- | --- |
| S.NO | Error Name | Error Rectification |
| 1 | Syntax/ Compilation Error | Absence of Semicolon |
| 2 | Closing Brackets | Need to Close the brackets |
| 3 | Class Name Error | Give the class name correctly |
| 4 | Constructor Calling | Call the constructor correctly |

Week-7

1. Write a Java program to create an abstract class Animal with an abstract method called sound(). Create subclasses Lion and Tiger that extend the Animal class and implement the sound() method to make a specific sound for each animal.

Code:

abstract class Animal {

abstract void sound();

}

class Lion extends Animal {

void sound() {

System.out.println("Lion roars: Roarrr!");

}

}

class Tiger extends Animal {

void sound() {

System.out.println("Tiger growls: Grrrr!");

}

}

public class AnimalSound {

public static void main(String[] args) {

System.out.println("Name :S. Sai Venkat"+"Roll No:AV.SC.U4CSE24322"+"Section:CSE B");

Lion l=new Lion();

l.sound();

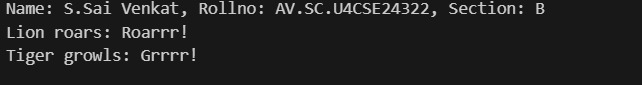
Tiger t=new Tiger();

t.sound();

}

}

Output:



Error Message:

|  |  |
| --- | --- |
| Error Message | Error Rectification |
| Java can’t find the main method to start your program Error rectification Convert static void main(String[ ] args) into | Convert static void main(String[ ] args) into Public static void main(String[ ] args |

Important points:

1)JVM(java virtual machine) is one which runs our program.JVM starts your program by calling the main( ) method from outside your class.JVM runs outside our package.only public method are accessible from outside a class or package.

2. )Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape.

CODE:

abstract class Shape3D{

abstract double CalculateVolume();

abstract double CalculateSurfaceArea();

}

class Sphere extends Shape3D{

double radius;

Sphere(double r) { this.radius=r;

}

double CalculateVolume() {

return (4.0/3.0)\*Math.PI\*radius\*radius\*radius;

}

double CalculateSurfaceArea() {

return 4\*Math.PI\*radius\*radius;

}

}

class Cube extends Shape3D{

double Side;

Cube(double s) {

this.Side=s;

}

double CalculateVolume() {

return Side\*Side\*Side;

}

double CalculateSurfaceArea() {

return 6\*Side\*Side;

}

}

public class ass18 {

public static void main(String[]args) {

Shape3D S=new Sphere(5.0); Shape3D C=new Cube(4.0);

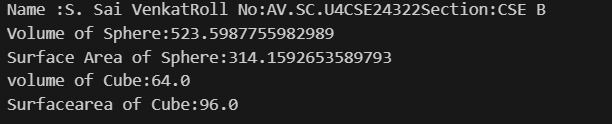
System.out.println("Name :S. Sai Venkat"+"Roll No:AV.SC.U4CSE24322"+"Section:CSE B");

System.out.println("Volume of Sphere:”+S.CalculateVolume()); System.out.println("Surface Area of Sphere:”+S.CalculateSurfaceArea()); System.out.println("volume of Cube:”+C.CalculateVolume()); System.out.println("Surfacearea of Cube:”+C.CalculateSurfaceArea());

}

}

Output:



Error message:

|  |  |
| --- | --- |
| Error message | Error rectification |
| Cannot find the symbol | Rectified by removing the Math.sPI by removing ‘s’in Math.sPI. |

3)write a java program using an abstract class to define a method for pattern printing

• Create an abstract class named patternprinter with an abstract method printpattern(int n) and a concrete method to display the pattern title

• Implement two subclasses

1. Star pattern prints a right angle triangle of Star( \*)

2. Numberpattern-prints a right angled triangle of increasing numbers

• In the main( ) method ,create objects of both subclasses and print the patterns for a given number of rows

Excepted Output:

Pattern 1:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

Pattern 2:

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

CODE:

abstract class PatternPrinter {

public abstract void printPattern(int n);

public void displayTitle(String title) {

System.out.println("\n" + title);

}

}

class StarPattern extends PatternPrinter {

@Override

public void printPattern(int n) {

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print("\* ");

}

System.out.println();

}

}

}

class NumberPattern extends PatternPrinter {

@Override

public void printPattern(int n) {

for (int i = 1; i <= n; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(j + " ");

}

System.out.println();

}

}

}

public class Pattern {

public static void main(String[] args) {

System.out.println("Name :S. Sai Venkat"+"Roll No:AV.SC.U4CSE24322"+"Section:CSE B");

int rows = 5;

PatternPrinter starPattern = new StarPattern();

starPattern.displayTitle("Pattern 1:");

starPattern.printPattern(row 1);

PatternPrinter numberPattern = new NumberPattern();

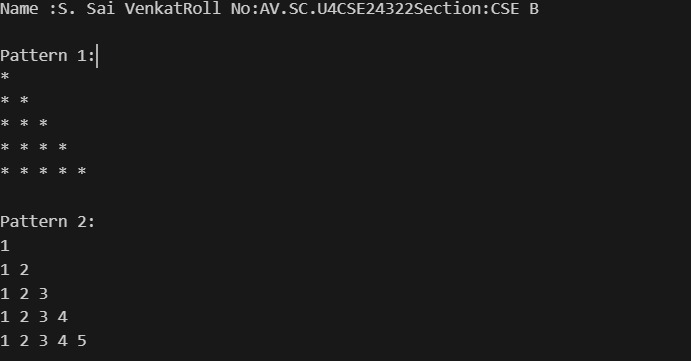
numberPattern.displayTitle("Pattern 2:");

numberPattern.printPattern(rows);

}

}

Output:



Error message:

|  |  |
| --- | --- |
| Error Message | Error Rectification |
| Iilegal start of expression | Rectified by mentioning the No.of rows i.e 5 rows. |

**WEEK-8**

**PROGRAM-1**

**AIM:**

Write a Java program to create an interface Shape with the getPerimeter() method. Create three classes Rectangle, Circle, and Triangle that implement the Shape interface. Implement the getPerimeter() method for each of the three classes.

Code:

class Shapes {

    public double area() {

        return 0;

    }

}

class Triangle extends Shapes {

    private double base;

    private double height;

    public Triangle(double base, double height) {

        this.base = base;

        this.height = height;

    }

    @Override

    public double area() {

        return 0.5 \* base \* height;

    }

}

class Circle extends Shapes {

    private double radius;

    public Circle(double radius) {

        this.radius = radius;

    }

    @Override

    public double area() {

        return Math.PI \* radius \* radius;

    }

}

class Rectangle extends Shapes {

    private double length;

    private double width;

    public Rectangle(double length, double width) {

        this.length = length;

        this.width = width;

    }

    @Override

    public double area() {

        return length \* width;

    }

}

public class ShapeArea {

    public static void main(String[] args) {

        Shapes triangle = new Triangle(2, 5);

        Shapes circle = new Circle(4);

        Shapes rectangle = new Rectangle(6, 9);

        System.out.println("Area of Triangle: " + triangle.area());

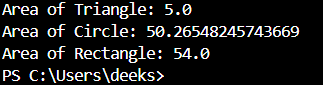
        System.out.println("Area of Circle: " + circle.area());

        System.out.println("Area of Rectangle: " + rectangle.area());

    }

}

Output:



CLASS DIAGRAM:

|  |
| --- |
| Shapes |
| +area(): double |

IMG_256

|  |
| --- |
| Triangle |
| -base                                    -height |
| IMG_257                                +area():double |

|  |
| --- |
| Circle |
| IMG_258                                       -radius |

|  |
| --- |
| Rectangle |
| -length                                   -width |
| +area():double |

Error table:

|  |  |
| --- | --- |
| IMG_259CODE ERROR      1)Class name "Shapes" is inconsistently used (should be consistent capitalization)    2)Base class method area() returns 0 by default - better to make it abstract | ERROR RECTIFICATION      Change to consistent capitalization (either all "Shapes" or all "Shapes")      2)Consider making Shapes abstract with abstract area() method |

       IMPORTANT POINTS:

***Inheritance Hierarchy***: The Traingle, Circle and Rectangle classes all inherit from the base Shapes class (note: class name is misspelled as "Shapes" in some places and "Shapes" in others).

***Polymorphism***: Each subclass overrides the area() method to provide its own implementation, demonstrating polymorphic behavior.

***Encapsulation***: All shape classes properly encapsulate their attributes (base, height, radius, length, width) as private fields.

***Method Overriding***: The area() method is overridden in each subclass with the appropriate calculation formula for that shape.

***Main Class***: The ShapeArea class demonstrates the use of these shapes by creating instances and calling their area() methods.

**PROGRAM-2:**

**AIM:**Write a Java program to create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.

interface Playable {

    void play();

}

class Football implements Playable {

    @Override

    public void play() {

        System.out.println("Playing Football: Kicking the ball towards the goal");

    }

}

class Volleyball implements Playable {

    @Override

    public void play() {

        System.out.println("Playing Volleyball: Bumping, setting, and spiking the ball");

    }

}

class Basketball implements Playable {

    @Override

    public void play() {

        System.out.println("Playing Basketball: Dribbling and shooting the ball");

    }

}

public class TestSports {

    public static void main(String[] args) {

        Playable football = new Football();

        Playable volleyball = new Volleyball();

        Playable basketball = new Basketball();

        football.play();

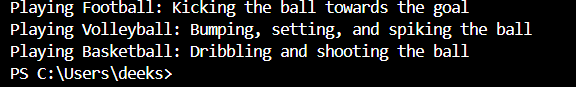
        volleyball.play();

        basketball.play();

    }

}

Output:



Class diagram:

|  |
| --- |
| Interface:playable |
| +play():void |

IMG_257

|  |
| --- |
| FootBall |
| +play():void |

IMG_258

|  |
| --- |
| VolleyBall |
| IMG_259                                       +play():void |

|  |
| --- |
| BasketBall |
| +play():void |

**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| Declaring an abstract class instead of interface class.  Not declaring public in each class. | Declare an interface class instead of abstract class.  Declare public infront of each class. |

**IMPORTANT POINTS:**

The playable interface abstracts the play() method, ensuring different classes implement it differently

The play() method behaves differently based on the object type football, volleyball, basketball.

Each class encapsulates its own implementation of how the sport is played, hiding the details from the user

**PROGRAM-3:**

**AIM:**Write a java program to implements login System using interfaces.

**CODE:**

interface LoginSystem {

    boolean login(String id, String pass);

}

class University\_portal implements LoginSystem {

    @Override

    public boolean login(String id, String pass) {

        if (id.equals("Student123") && pass.equals("pass02")) {

            System.out.println("Login successful");

            return true;

        } else {

            System.out.println("Invalid credentials");

            return false;

        }

    }

    public static void main(String[] args) {

        University\_portal p1 = new University\_portal();

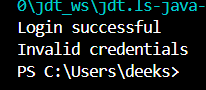
        p1.login("Student123", "pass02");

        p1.login("Student123", "wrongpass");

    }

}

Output:



ERROR TABLE:

|  |  |
| --- | --- |
| IMG_261 CODE ERROR    Saving file with the interface name is error  Removing public will leads to error | ERROR RECTIFICATION    save with implement name.    add public static void main  (String[]args) |

**IMPORTANT POINTS:**

***Interface Implementation:***

University\_potral correctly implements Login System interface

Uses Override annotation for the login() method

***Authentication Logic***:

Hardcoded credentials: id="Student123", password="pass02"

Returns boolean and prints appropriate message.

***Main Method***:

Demonstrates both successful and failed login attempts

Includes student information print statement

***Polymorphism***:

Could create Login System Portal=new University\_portal();

Demonstrates interface-based programming