23CSE111

OBJECT-ORIENTED PROGRAMMING

LAB REPORT



Department of Computer Science and Engineering

Amrita School of Engineering

Amrita Vishwa Vidyapeetham, Amaravati Campus

Verified by: NAME: B.keerthi

ROLL NO:AV.SC.U4CSE240

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Title** | Date | Page No. | Signature |
| Week 1 |  | 27-01-2025 |  |  |
| 1. | How to download and install Java Software. |  |  |  |
| 2. | Write a Java Program to print the message “Welcome to Java Programming”. |  |  |  |
| 3. | Write a Java program that prints: Name, Roll.no. section of a student. |  |  |  |
| Week 2 |  | 10-02-2025 |  |  |
| 1. | Write a java program to calculate the area of a rectangle. |  |  |  |
| 2. | Write a java program to temperature from Celsius to Fahrenheit and vice-versa. |  |  |  |
| 3. | Write a java program to calculate the simple interest. |  |  |  |
| 4. | Write a java program to find the largest of three numbers, using ternary operator. |  |  |  |
| 5. | Write a java program to find the factorial of a number. |  |  |  |
| Week 3 |  | 24-02-2025 |  |  |
|  | To create a java program with the following instructions:   1. Create a class name “car” 2. Create 4 attributes named car\_color, car\_brand, fuel\_type and mileage 3. Create 3 methods names start (), stop () and service (). 4. Create 3 objects named car1, car2 and car3 5. Create a constructor which should print “welcome to car garage”. |  |  |  |
|  | To write a java program to create a class named BankAccount, with 2 methods deposit () and withdraw ().   1. Deposit (): Whenever an amount is deposited, it has to be updated the current amount. 2. Withdraw (): Whenever an amount is withdrawn, it has to be less than the current amount, else print (“Insufficient funds”). |  |  |  |
| Week 4 |  | 03-03-2025 |  |  |
| 1. | Write a java program with class named “book”. The class should contain various attributes such as “title\_of\_the\_book”, “author”, “year\_of\_publication”. It should also contain a constructor with the parameters which initializes “title\_of\_the\_book”, “author”, “year\_of\_publication”. Create a method which displays the details of the book, that is “author”, “title\_of\_the\_book”, “year\_of\_publication”. Display the details of two books by creating two objects. |  |  |  |
| 2. | To create a java program with class name “my\_class” with a static variable “count” of “int” type, initialized to zero and a constant variable “pi” of type double initialized to “3.1415” as attributes of that class. Define a constructor for “my\_class” that increments the count variable each time an object of my\_class is created. Finally print the values of “count” and “pi” |  |  |  |

WEEK 1

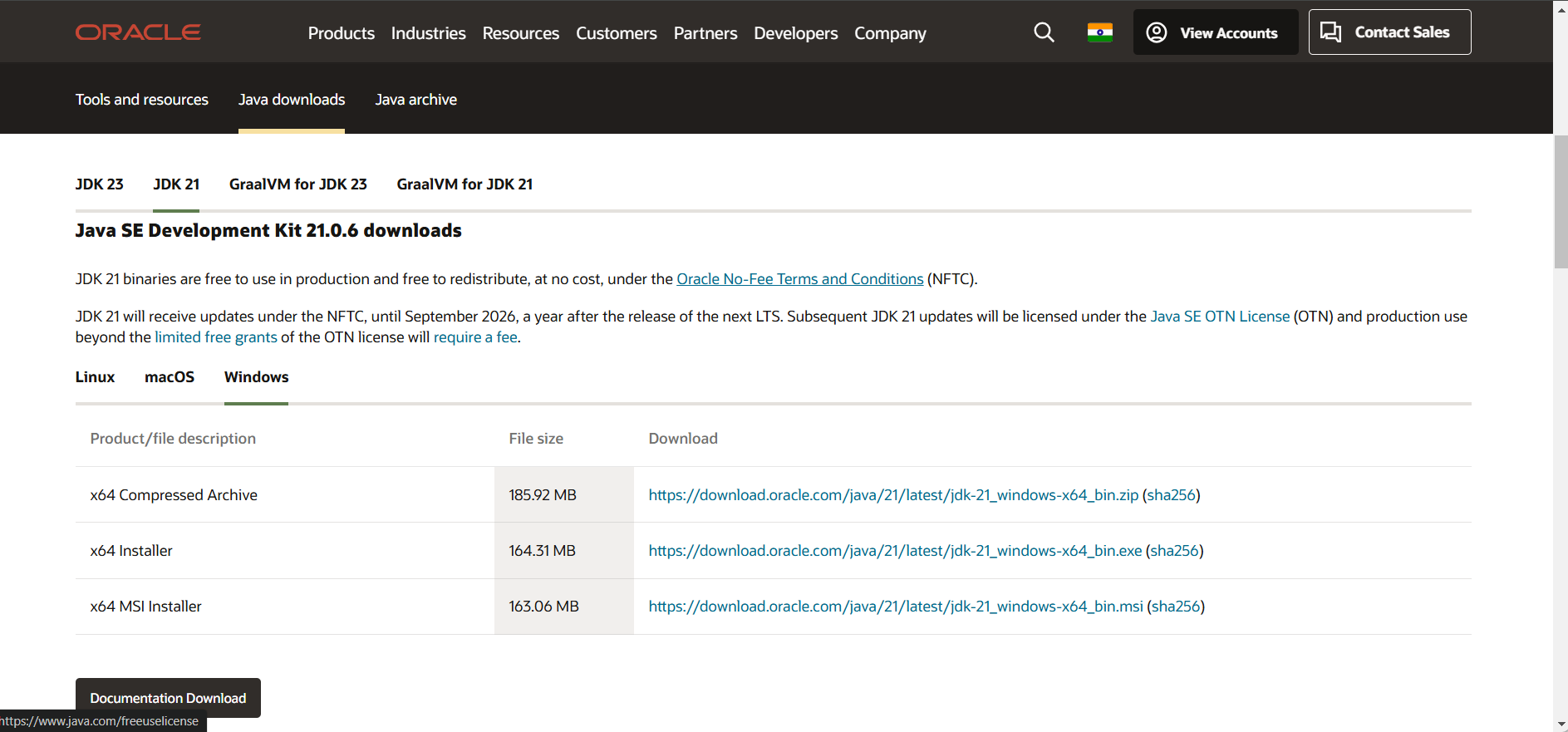
PROGRAM – 1

AIM: How to download and install java software

PROCEDURE: The procedure for downloading the software is written below.

Step 1: Downloading JDK21

1. Open your default web browser and go to the search bar and type Oracle JDK Downloads page.
2. Scroll down and click on Java SE Development Kit21 section.
3. Choose “Windows x64” Installer version.
4. Click on Downloads and wait for the whole installation to get completed.

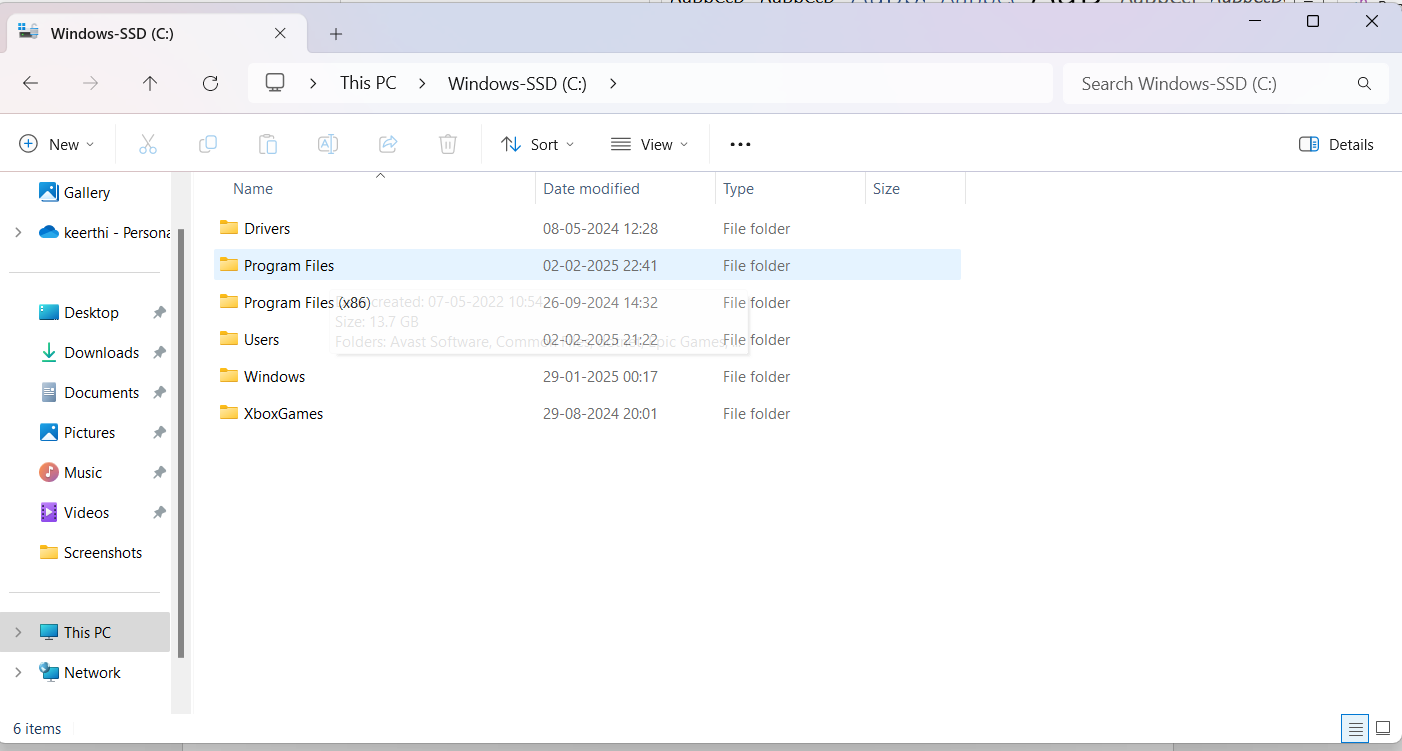


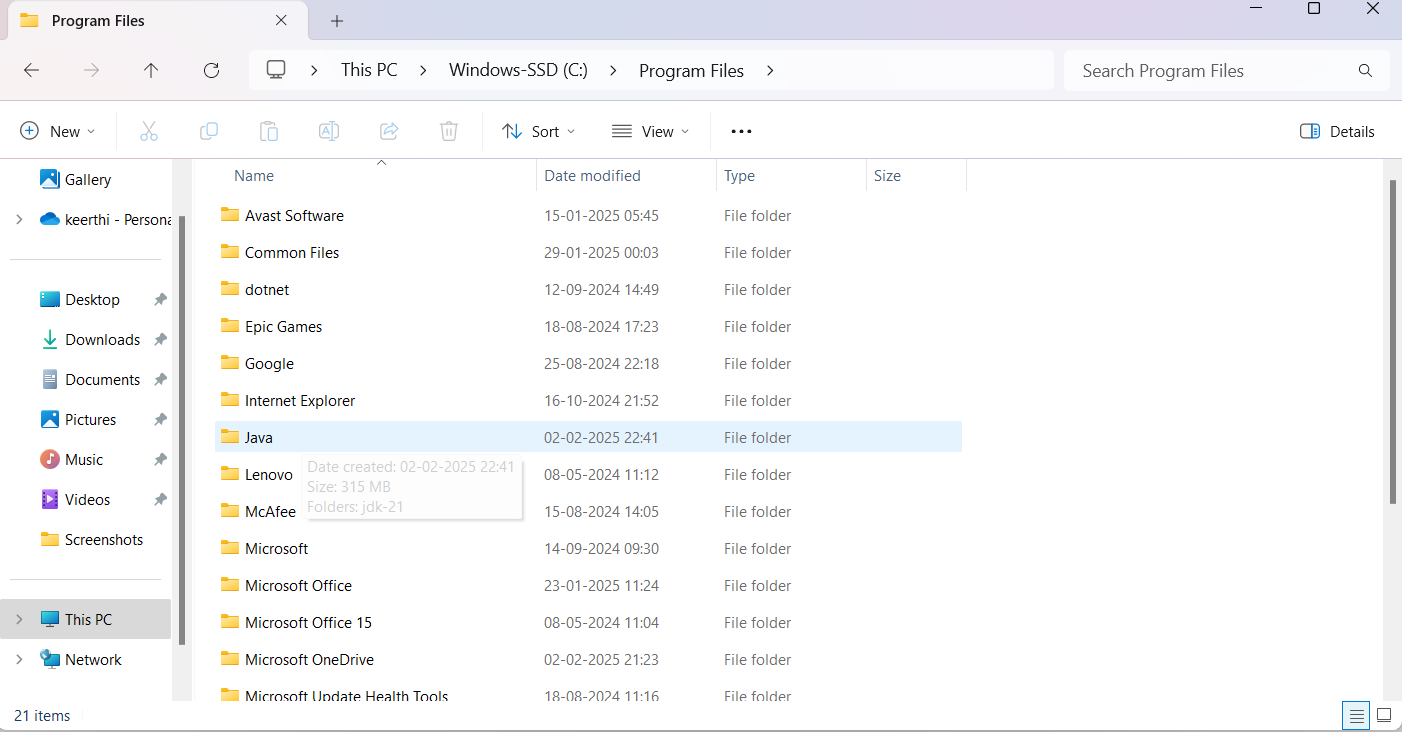
Step 2: Installing of JDK21

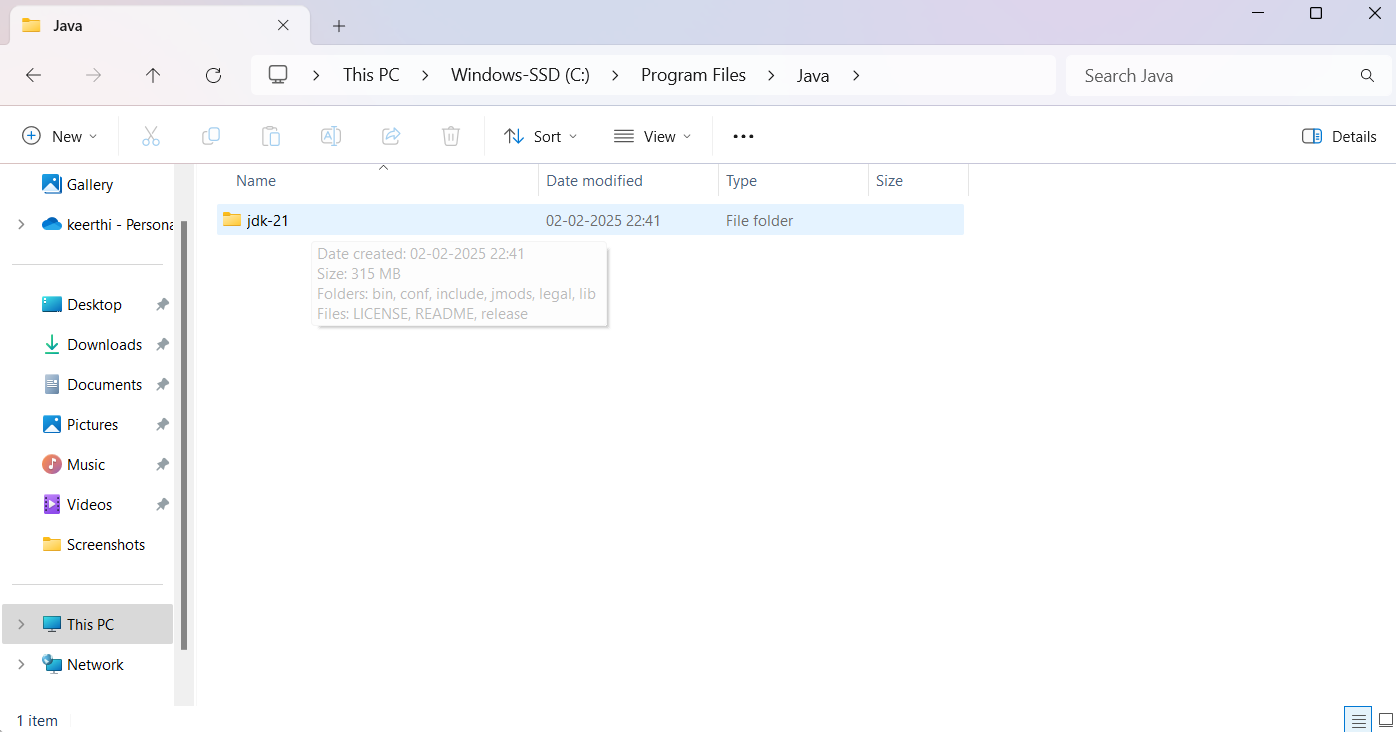
1. Locate the downloaded document jdk-21\_windows-x64\_bin.exe file.
2. Double-click on it and launch the installer.
3. Click on the next simultaneous steps.
4. Choose the installation path (C:/ProgramFiles/java/jdk-21).
5. Click on the next step and install it.
6. Wait for the process of installation to get completed.
7. Click on the close button and the whole installation is completed.

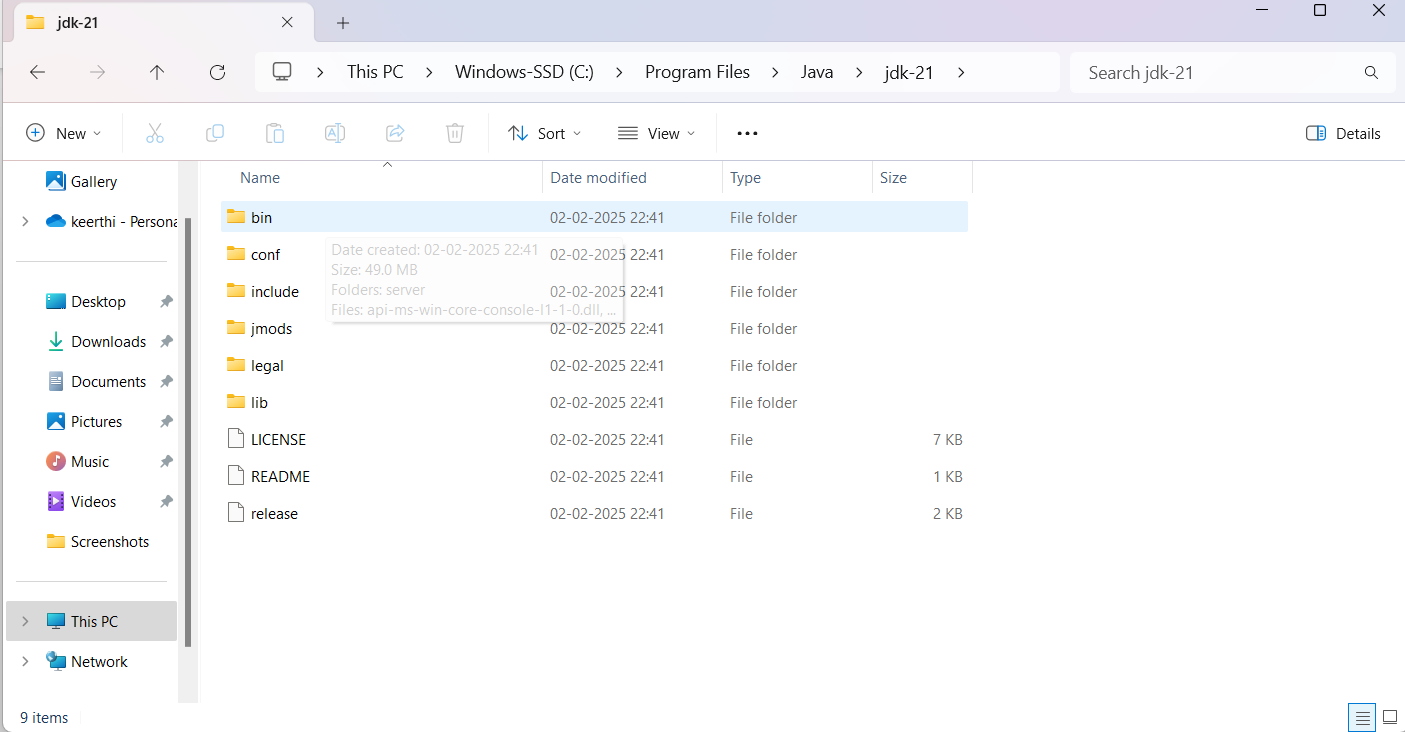
Step 3: Setting up the path of Java program

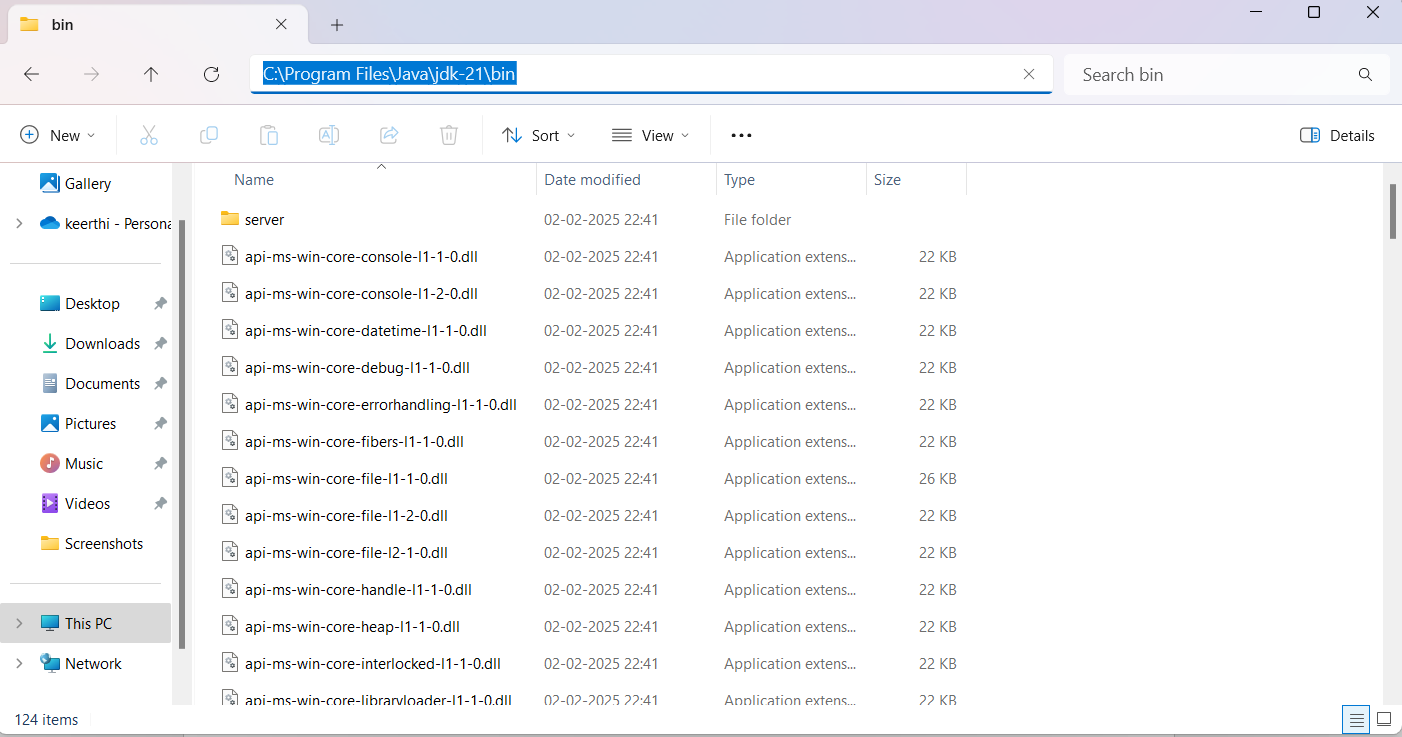
1. Access the “windows c: drive” on your laptop.
2. Choose the program files option and click on java, then click on JDK version 21, and then select the bin option.
3. Copy the path address at the top.





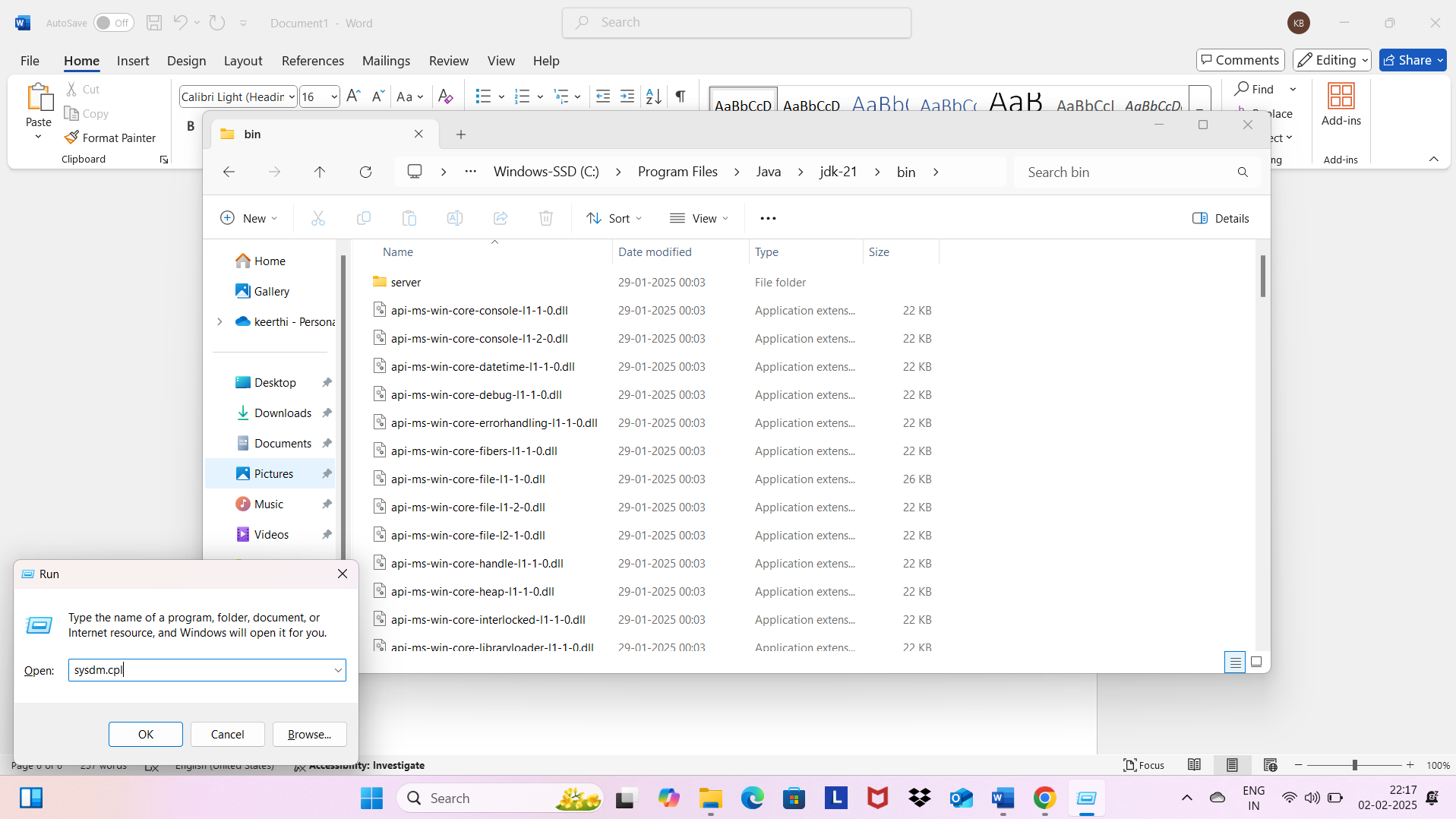


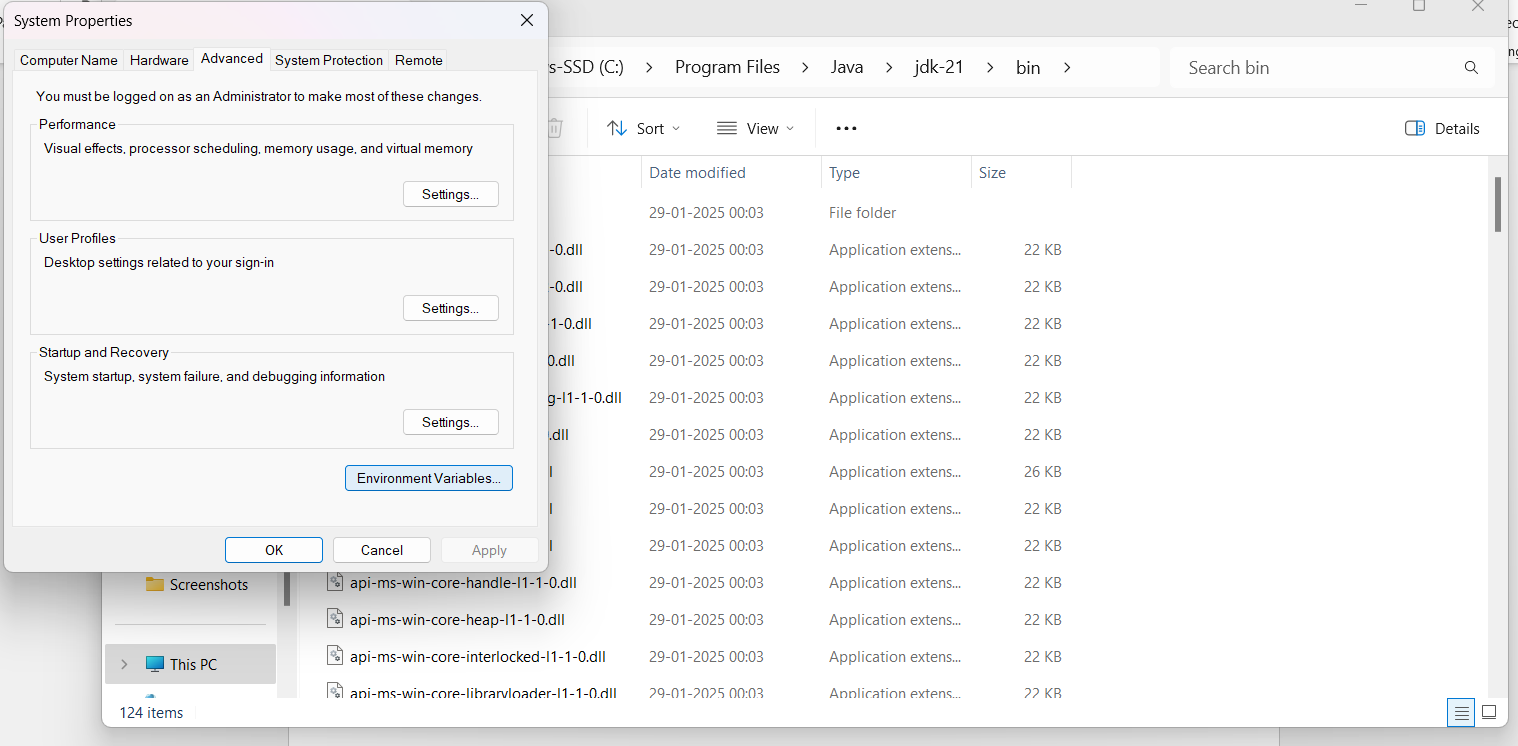




Step 4: Accessing the system properties.

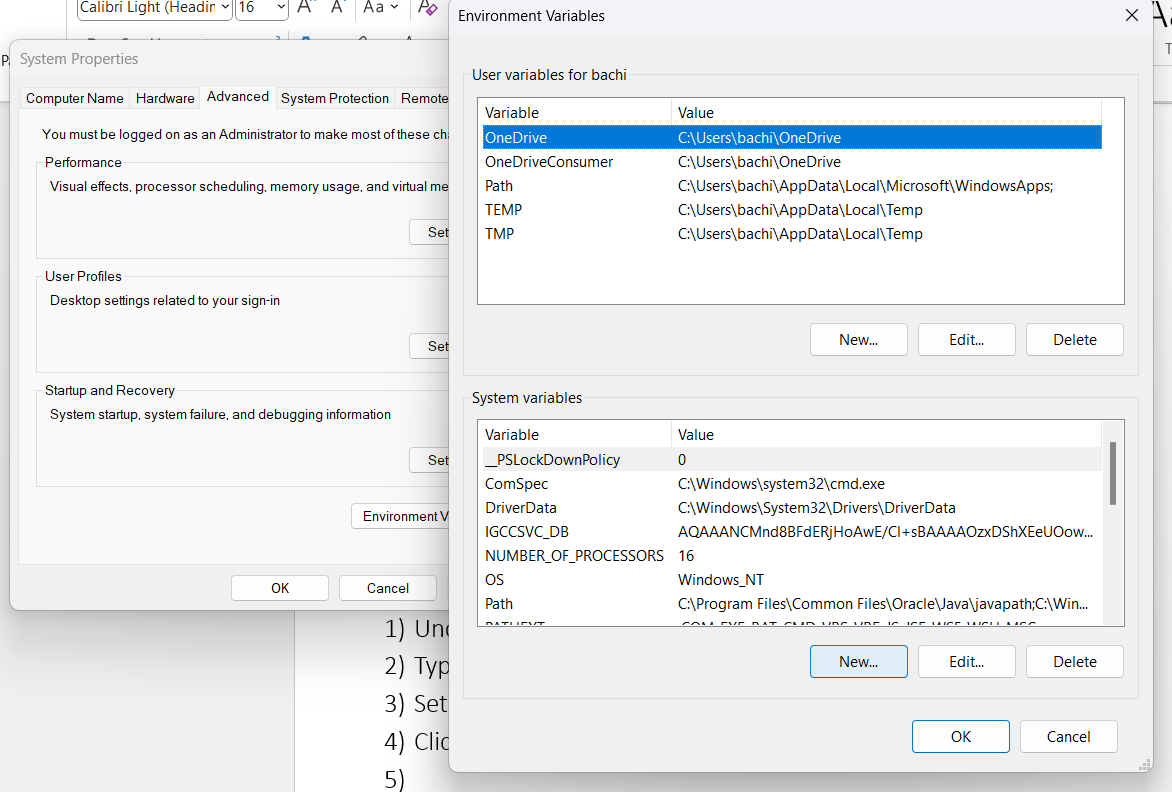
1. Press the windows + R option and type sysdm.cpl, and click on the option “ok”.
2. The system properties will open.
3. Then click on the Advanced tab.
4. Go to the environment variables present at the bottom.

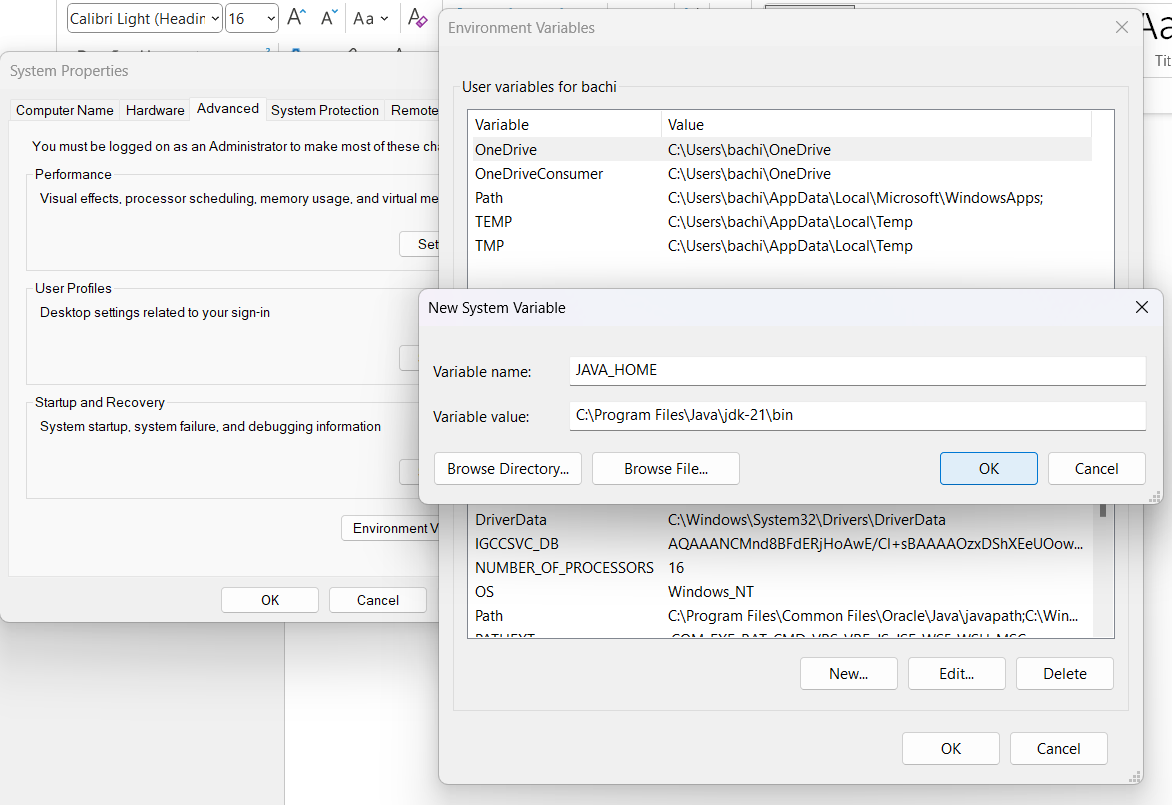




Step 5: Setting up JAVA\_HOME

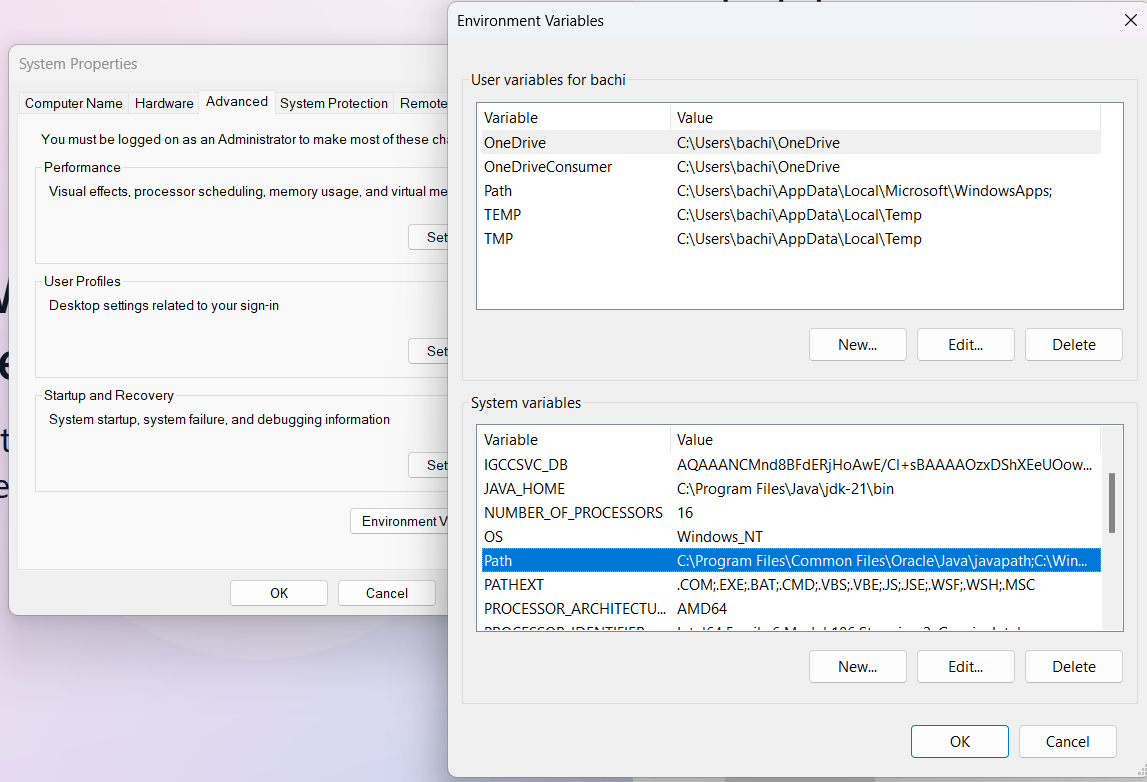
1. Under the system variables option click on the option “new”.
2. Type out the name as JAVA.HOME.
3. Set the variable value as the installation path.
4. Click on the function “ok”.

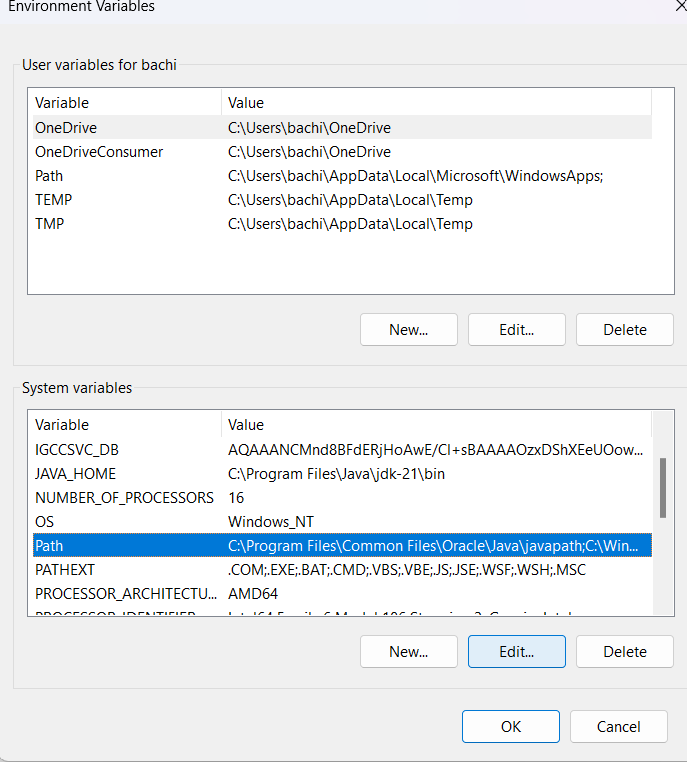
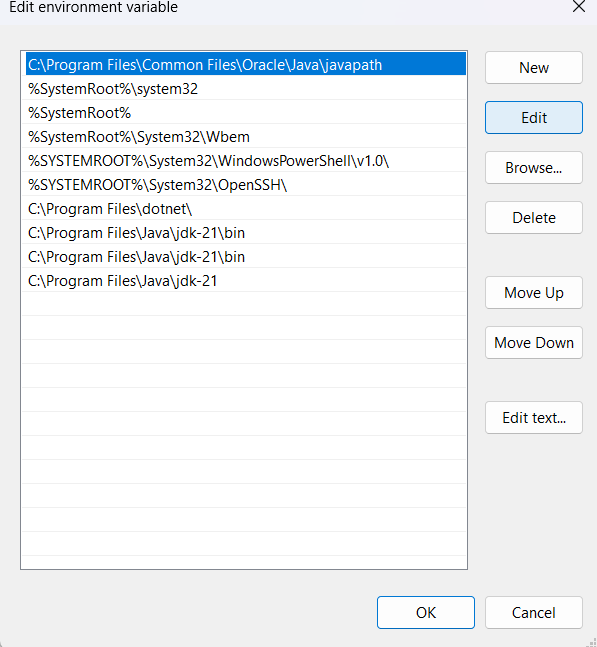


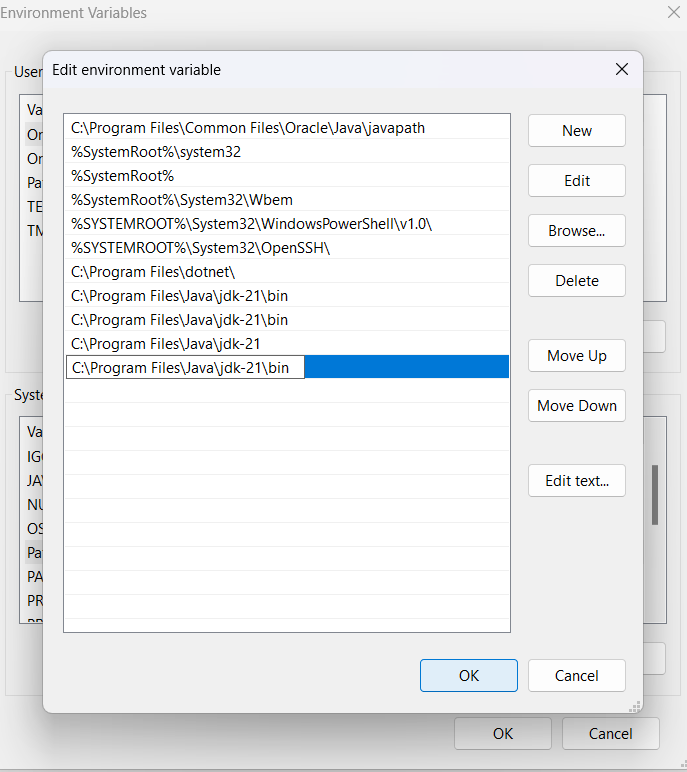


Step 6: Updating the PATH variable.

1. In the system variable option, find the path and click on edit option.
2. Click on the new option and add C:\Program Files\Java\jdk-21\bin.
3. Click on OK to save.

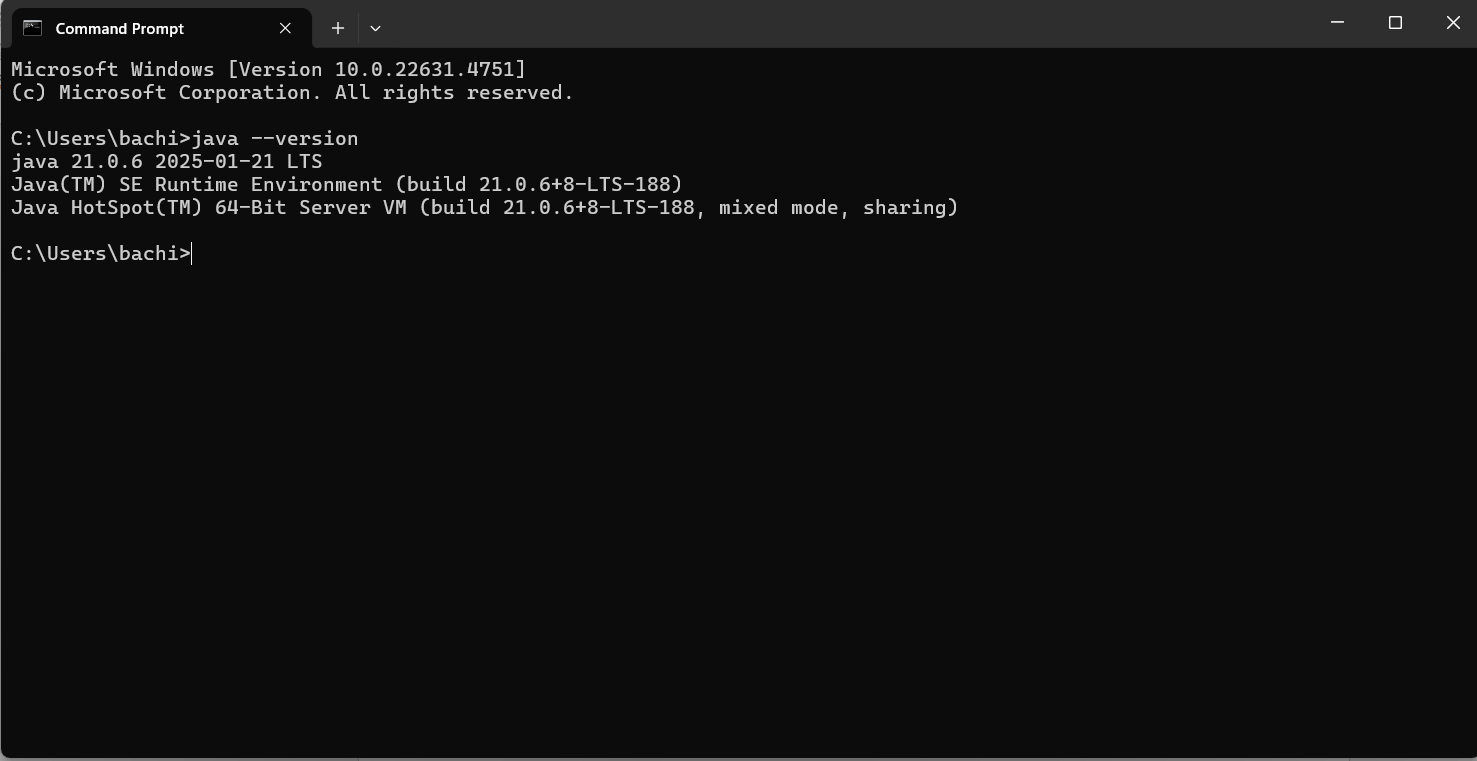


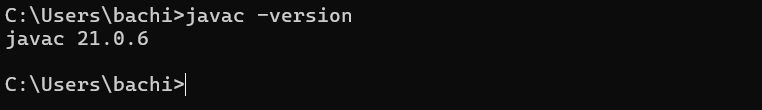


Step 7: verifying the process of installation.

1. Click on the command prompt in your system.
2. Type out the command, **java --version** and click on enter.



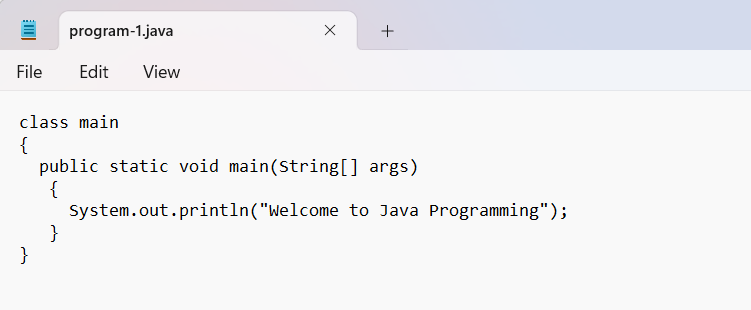
1. We can also check the compiler version by typing out the following command.



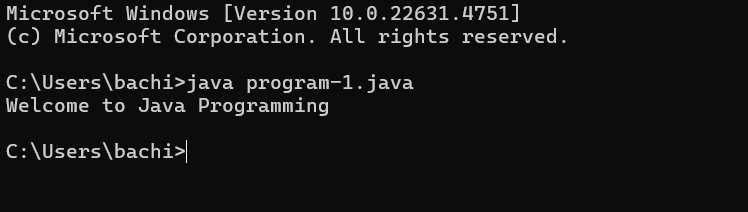
PROGRAM 2:

AIM: To write a java program and print the message “Welcome to Java Programming”.

INPUT:



OUTPUT:

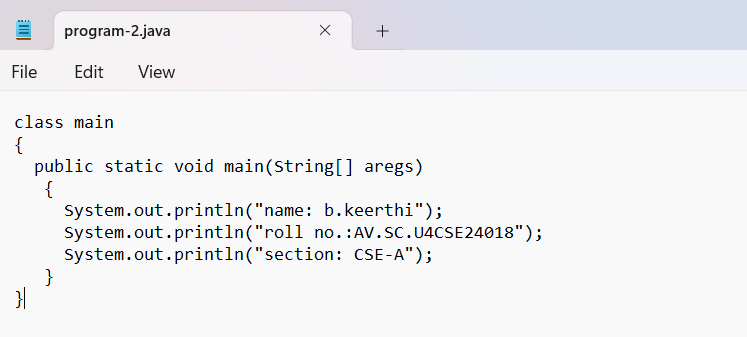


ERRORS: No errors were to be found.

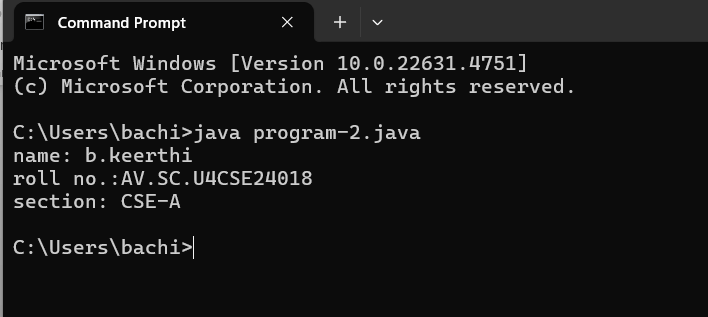
PROGRAM 3:

AIM: Writing of a java program to print out name, roll no., and section of a student.

INPUT:



OUTPUT:



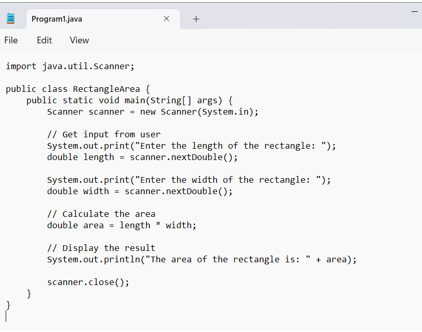
ERRORS: No errors were found as of the program.

WEEK 2

PROGRAM – 1:

AIM: write a java program to calculate the area of the rectangle.

INPUT:



ERRORS:

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Error message** | **Error rectification** |
| 1. | Error: <identifier> expected. Public static void main. Compilation failed | Public Static void main (String [ ] args) |

OUTPUT:



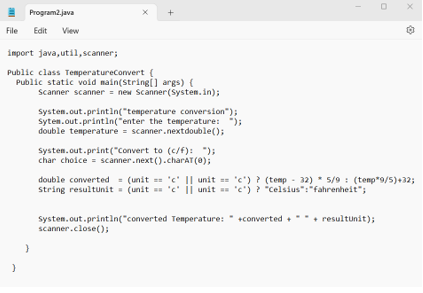
CONCEPTS KNOWN:

* Import java.util.Scanner – used to accept inputs from the user, under the util package has to be imported.
* Scanner input=new Scanner(System.in); - Used to create a Scanner object
* int ln=input.nextInt(); - Used to read the integer data type stored under the object created
* System.out.println(“ “); - It is used to print string inside the quotes. After printing, the cursor moves to the beginning of the next line.

PROGRAM 2

AIM: Write a java program to convert temperature from Celsius to Fahrenheit and vice-versa.

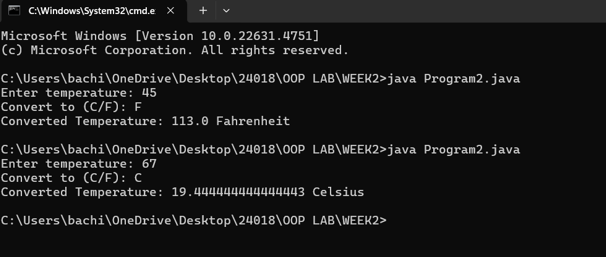
INPUT:



ERRORS:

|  |  |  |
| --- | --- | --- |
| **SNO.** | **Error message** | **Error rectification** |
| 1. | Error: ‘. ‘expected | Remove the ‘ , ‘ symbol and replace with the correct symbol |
| 2. | Error: class, interface, enum, or record expected | Static is written correctly |
| 3. | Error: <identifier> expected | Static is written correctly |

OUTPUT:



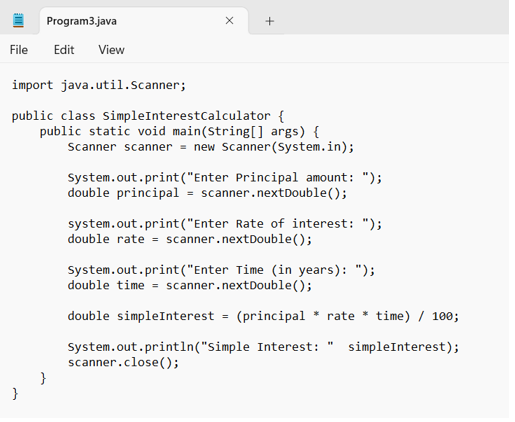
CONCEPTS KNOWN:

* import java. util. Scanner; - To accept input from user, Scanner class under util package has to be imported.
* Scanner input=new Scanner (System.in); - Used to create a Scanner object
* double fh=input. next Double (); - Used to read double data type stored under the object created
* System.out.println(““); - It is used to print string inside the quotes. After printing, the cursor moves to the beginning of the next line.

PROGRAM-3

AIM: Write a java program to calculate the simple interest.

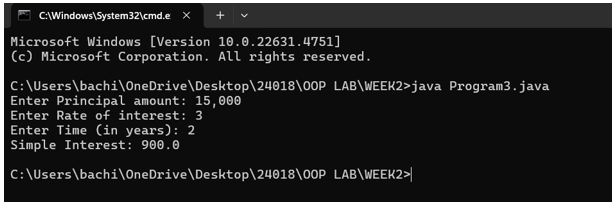
INPUT:



ERRORS:

|  |  |  |
| --- | --- | --- |
| **SNO** | **Error message** | **Error rectification** |
| 1. | Error: ‘) ‘or ‘, ‘is expected | Error rectified at line 5 |
| 2. | Error: not a statement | Statement has been verified at line 5 |
| 3. | Error: ‘; ‘is expected | Error rectified at line 6 |

OUTPUT:



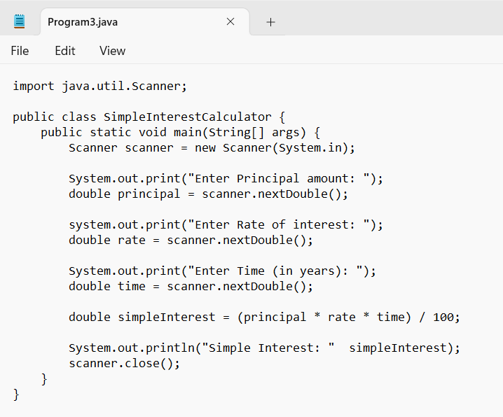
CONCEPTS TO KNOW:

* import java. util.Scanner; - To accept input from user, Scanner class under util package has to be imported.
* Scanner input=new Scanner (System.in); - Used to create a Scanner object
* double p=input. Next Double (); - Used to read double data type stored under the object created
* System.out.println (““); - It is used to print string inside the quotes. After printing, the cursor moves to the beginning of the next line.

PROGRAM 4:

AIM: Write a java program to find the largest of three numbers, using ternary operator

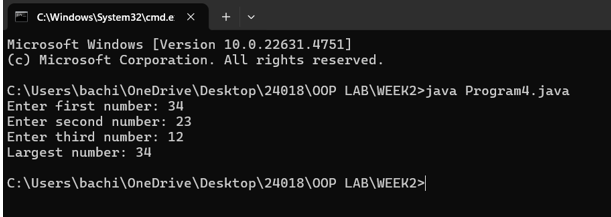
INPUT:



ERRORS:

|  |  |  |
| --- | --- | --- |
| **SNO** | **Error message** | **Error rectification** |
| 1. | Error: package ava.util does not exist | Error rectified and changed to java. util |
| 2. | Error: cannot find the symbol | A symbol “= “which is missed is written again. |
| 3. | Error: cannot find the symbol | Missed symbol is replaced. |
| 4. | Error: package system does not exist | A symbol ‘. ‘Which has been missed is written. |
| 5. | Error: cannot find symbol | The symbol ‘; ‘is written |

OUTPUT:



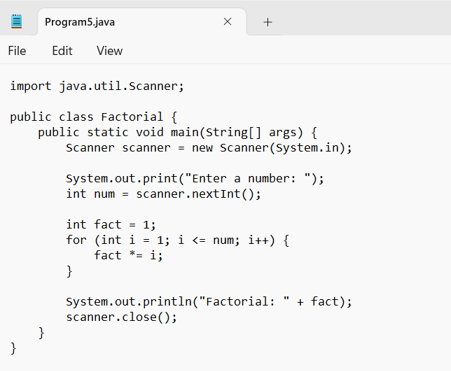
CONCEPTS KNOWN:

* import java.util.Scanner; - To accept input from user, Scanner class under util package has to be imported.
* Scanner input=new Scanner(System.in); - Used to create a Scanner object
* int a=input.nextInt (); - Used to read integer data type stored under the object created
* int result=(a>b)? ((a>c)? a:c): ((b>c)? b:c); - Nested Ternary operator is used here.
* Syntax for ternary operator is- condition? expression 1: expression 2; whose answer is stored in a variable and then used.

PROGRAM 5:

AIM: Write a java program to find the factorial of a number.

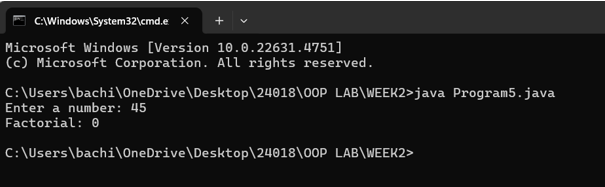
INPUT:



ERRORS:

|  |  |  |
| --- | --- | --- |
| **SNO.** | **Error message** | **Error rectification.** |
| 1. | Error: class, interface, enum, or record is expected. | The word factorial has been replaced |
| 2. | Error: unnamed classes are a preview feature and are disabled by default. | The word “static” has been replaced with “Static” |
| 3. | Error: class, interface, enum, or record expected | The symbol ‘} ‘has been removed. |

OUTPUT:



CONCEPTS KNOWN:

* for (int i=1; n>=i; --n) {} - For loop syntax: for (initial expression; test expression; update expression {} The loop is executed, until the test expression evaluates to be false

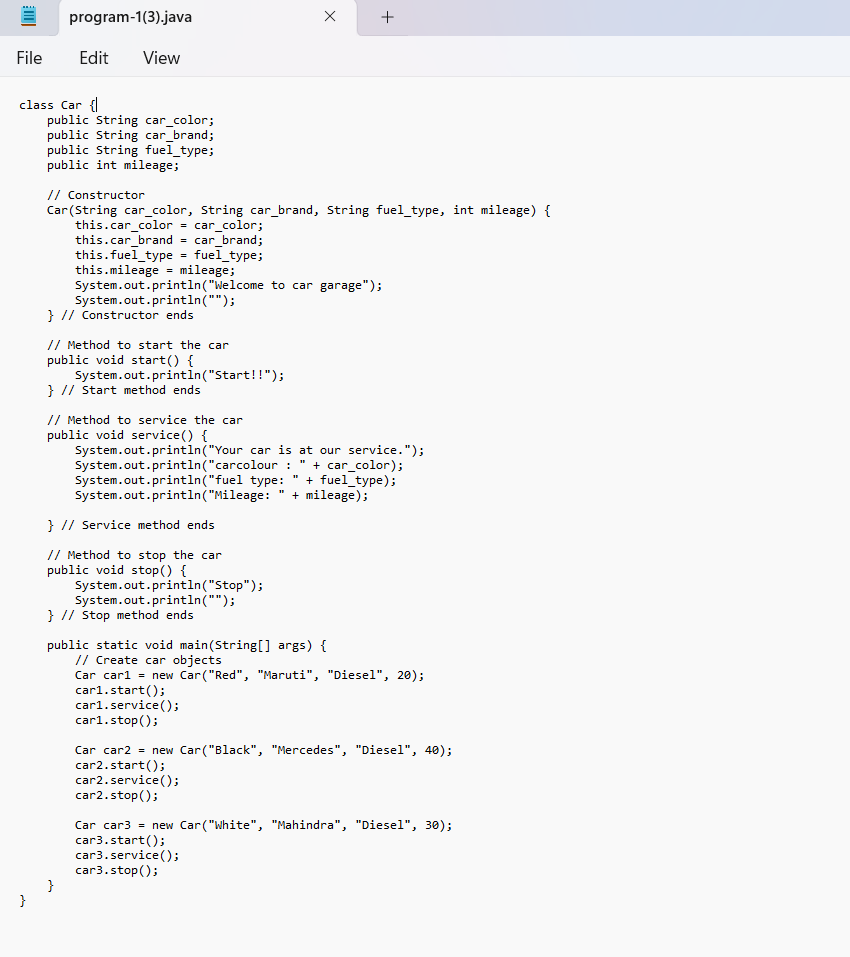
WEEK – 3

PROGRAM 1:

AIM: To create a java program with the following instructions:

1. Create a class with name “Car”
2. Create 4 attributes, named: car\_color, car\_brand, fuel\_type, mileage
3. Create 3 methods, named: start (), service (), stop ()
4. Create 3 objects, named: car1, car2, car3
5. Create a constructor, which should print, “Welcome to car garage”.

INPUT:



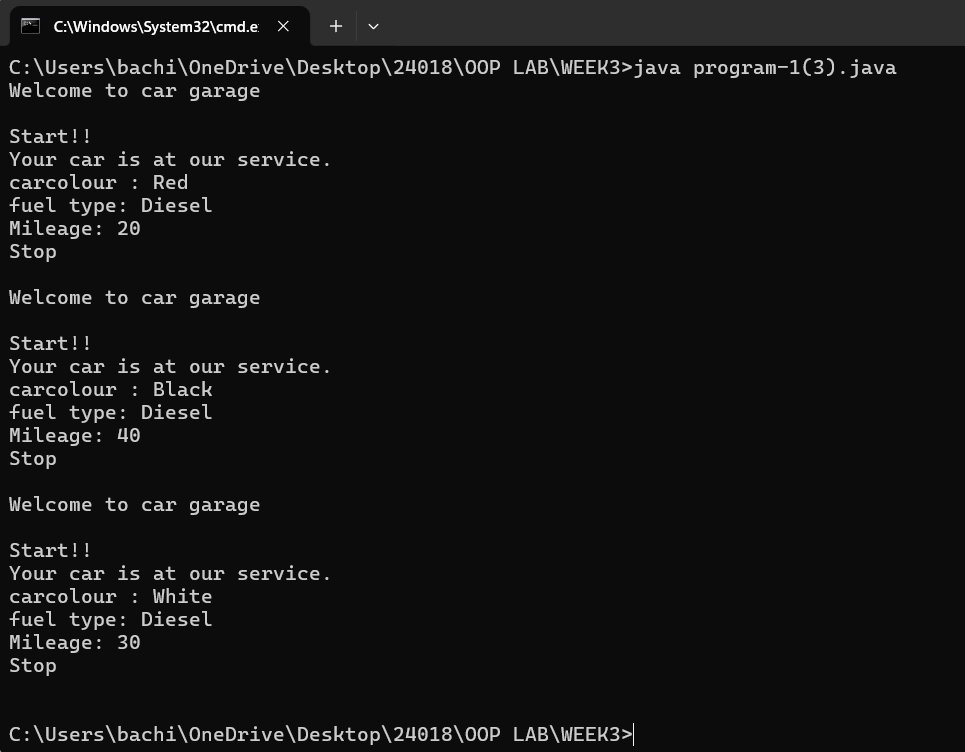
ERRORS:

|  |  |  |
| --- | --- | --- |
| Sno. | Error message | Error rectification |
| 1. | error: ';' expected car1.start() | Add a “;”  car1.start(); |
| 2. | error: illegal start of type  public void stop () | Add a “)”  public void stop () { |
| 3. | error: cannot find symbol thiscar\_brand=car\_brand; | Add a “.”  this.car\_brand=car\_brand; |

CLASS DIAGRAM:

|  |
| --- |
| Car |
| + car\_color: String  + car\_brand: String  + fuel\_type: String  + mileage: int |
| + Car (): void  + start (): void  + service (): void  + stop (): void |

OUTPUT:



CONCEPTS KNOWN:

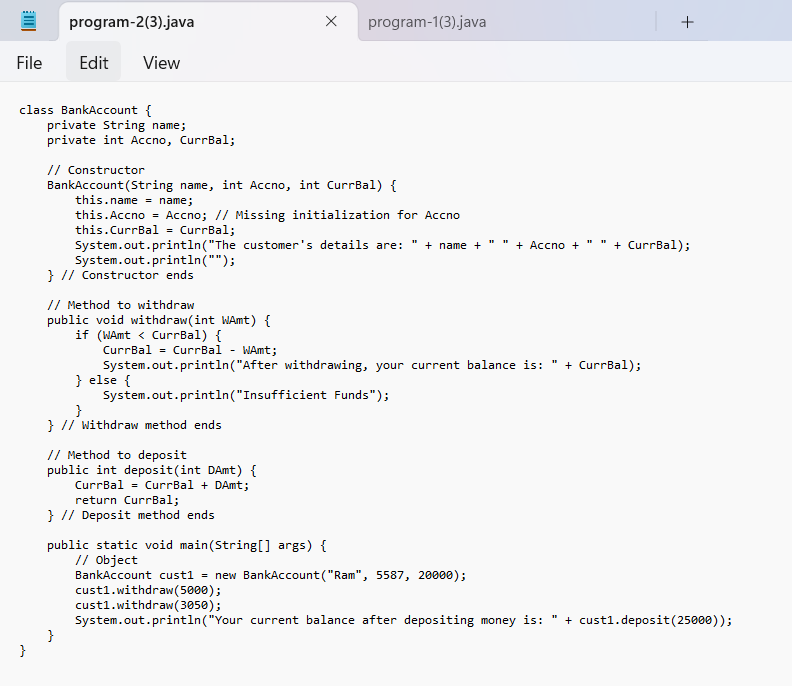
* public String car\_color; - Used to declare a variable named car\_color, with data type as String with public accessibility.
* Car (String car\_color, String car\_brand, String fuel\_type, int mileage) { } – It is a constructor (method with name same as class), which requires parameters such as car\_color (String data-type) and so on.
* this.car\_color=car\_color; - “this” is a default method, which is used to point to the instance variables.
* public void start () {} – used to declare a method, which will return nothing(void) in public accessibility.

PROGRAM 2:

AIM: To write a java program to create a class named Bank Account, with 2 methods deposit () and withdraw ().

1. Deposit (): Whenever an amount is deposited, it has to be updated with the current amount.
2. Withdraw (): Whenever an amount is withdrawn, it has to be less than the current amount, else print (“Insufficient funds”)

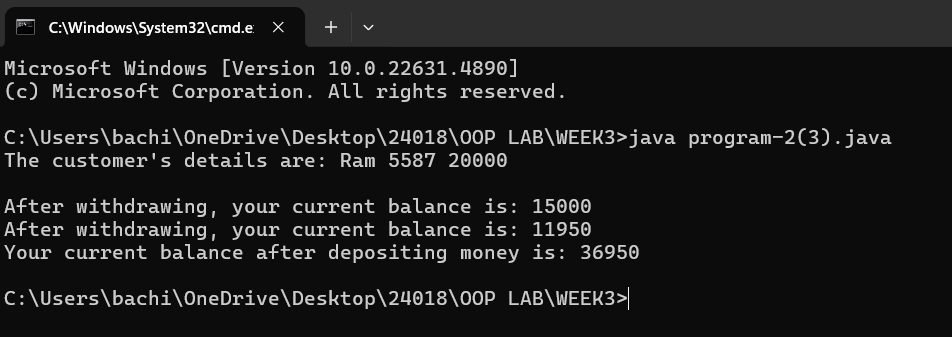
INPUT:



ERRORS:

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Error message** | **Error rectification** |
| 1. | error: ';' expected cust1.withdraw(3050) | Add a “;”  cust1.withdraw(3050); |
| 2. | error: cannot find symbol thisCurrBal=CurrBal; | Add a “.”  this.CurrBal=CurrBal; |

OUTPUT:



CLASS DIAGRAM:

|  |
| --- |
| BankAccount |
| - name: String  - Accno: int  - CurrBal: int |
|  |

CONCEPTS KNOWN:

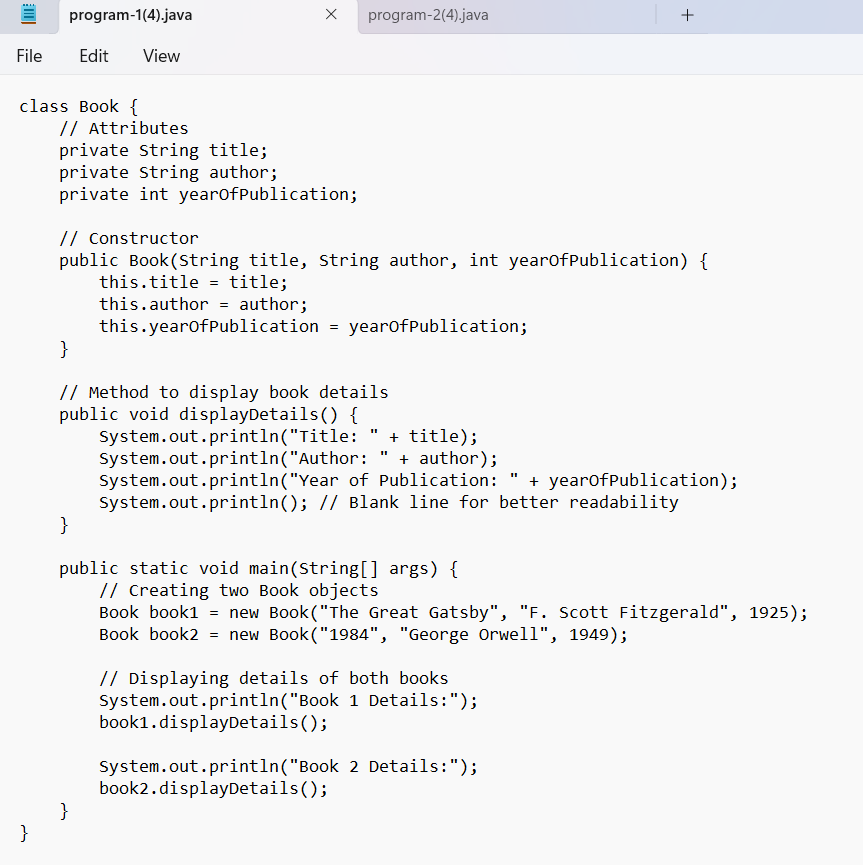
* private String name; - Used to declare a variable named name, with data type as String with private accessibility.
* BankAccount (String name, int Accno, int CurrBal) {} – It is a constructor (method with name same as class), which requires parameters such as name (String data-type) and so on.
* this.CurrBal=CurrBal; - “this” is a default method, which is used to point to the instance variables.
* public void withdraws (int WAmt) {} – used to declare a method, which will return nothing(void) in public accessibility, which requires a parameter WAmt (integer data type).
* public int deposit (int DAmt) {} - used to declare a method, which will return integer data type in public accessibility, which requires a parameter DAmt (integer data type).
* BankAccount cust1=new BankAccount("Ram",5587,20000); - used to create an object in class BankAccount, with object name as cust1.
* cust1.withdraw(50000); - Calling a method, under object cust1, by passing a parameter.
* System.out.println("Your current balance after depositing money is:"+cust1.deposit(25000)); - Deposit method will return the value, which will be directly printed

WEEK-4

PROGRAM-1:

AIM: Write a java program with class named “book”. The class should contain various attributes such as “title\_of\_the\_book”, “author”, “year\_of\_publication”. It should also contain a constructor with the parameters which initializes “title\_of\_the\_book”, “author”, “year\_of\_publication”. Create a method which displays the details of the book, that is “author”, “title\_of\_the\_book”, “year\_of\_publication”. Display the details of two books by creating two objects.

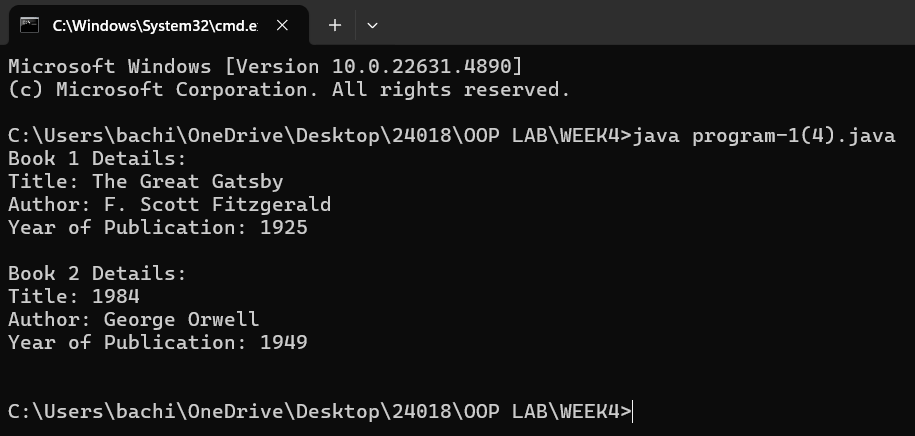
INPUT:

****

**ERRORS:**

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Error message** | **Error rectification** |
| 1. | Error: ‘;’ is expected | In line no.16 the required symbol is added. |
| 2. | Error: cannot find the symbol | Line 5 string changed to int and S is capital is string. |
| 3. | Error: ‘)’ and ‘;’ is expected | The space at the last line is removed. |
| 4. | Error: class, interface, enum or record expected | Additional symbol ‘{‘ is removed at the end. |

OUTPUT:



CLASS DIAGRAM:

|  |
| --- |
| Book |
| - title: String  - author: String  - yearOfPublication: int |
| + Book(title, author, yearOfPublication)  + displayDetails(): void |

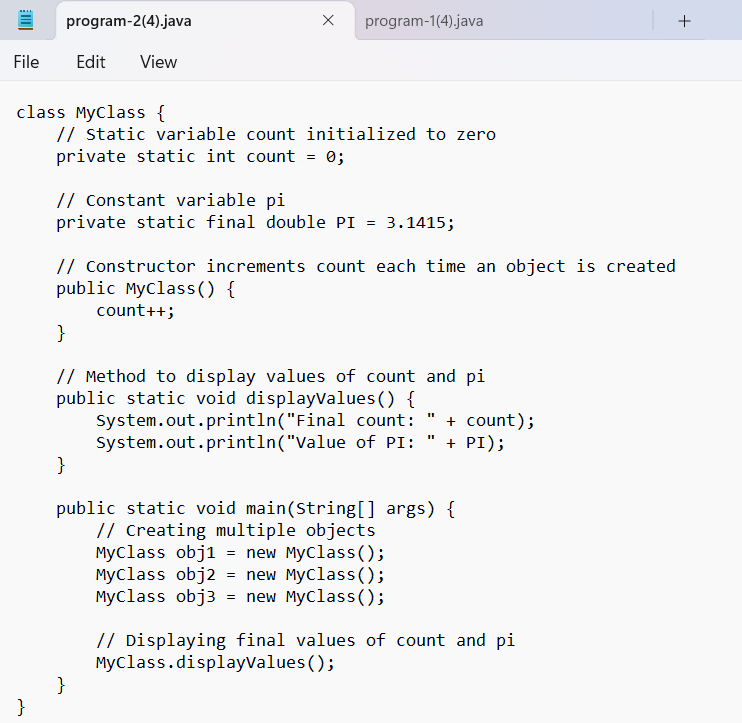
CONCEPTS KNOWN:

* The Book class defines the attributes and behavior of a book.
* Objects (book1 and book2) are created from the Book class.
* The attributes (title, author, yearOfPublication) are declared private to restrict direct access.
* Getter and setter methods could be added to allow controlled access.
* Two objects, book1 and book2, are created using the new keyword.
* The System.out.println() function is used to display book details.

PROGRAM-2

AIM: To create a java program with class name “my\_class” with a static variable “count” of “int” type, initialized to zero and a constant variable “pi” of type double initialized to “3.1415” as attributes of that class. Define a constructor for “my\_class” that increments the count variable each time an object of my\_class is created. Finally print the values of “count” and “pi”.

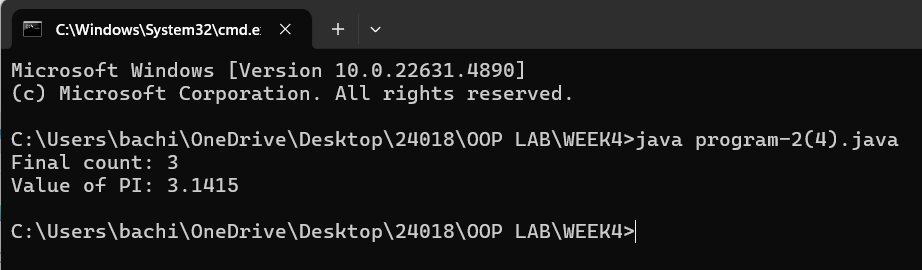
INPUT:



ERRORS:

|  |  |  |
| --- | --- | --- |
| **SNO.** | **Error message** | **Error rectification** |
| 1. | Error: non-static variable count cannot be referenced | Change count++ to my\_class.count++ inside the constructor. |
| 2. | Error: non-static method display() cannot be referenced | Make display() a static method or call display() using an object instance. |
| 3. | Error: non-static variable pi cannot be referenced | Use an instance of my\_class to access pi, like obj1.pi. |

OUTPUT:



CLASS DIAGRAM:

|  |
| --- |
| Myclass |
| - count: int  - pi: double |
| + Myclass()  + display(): void |

CONCEPTS KNOWN:

* static int count is shared among all instances of my\_class.
* It is incremented every time an object is created.
* final ensures that pi remains constant throughout execution.
* The constructor my\_class() increases count each time an object is instantiated.
* This demonstrates object creation and constructor execution.