**23CSE111**

**Object Oriented Programming**



**Department of Computer Science Engineering Amrita School of Engineering**

**Amrita Vishwa Vidyapeetham, Amaravati Campus**

**Name: Mehuli Sarkar**

**Verified By Roll No:AV.SC.CSE24150**

|  |  |  |  |
| --- | --- | --- | --- |
| S No. | Questions | Page No | Signature |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |

**Week-1**

**JAVA:**

Java is a high level, class based, object oriented programming language that is widely used across various operating systems.

**Installation of JAVA in Computer:**

**Aim:**

Solving computational programming using JAVA.

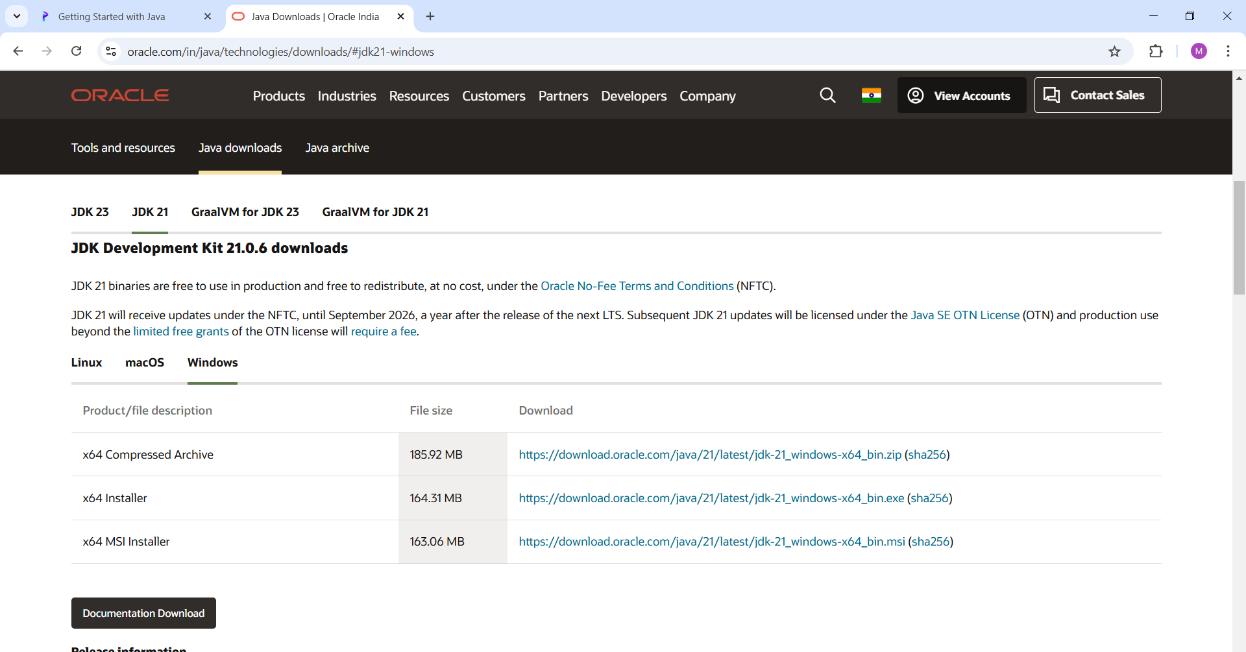
**Procedure:**

Follow the steps below to install Java on Windows:

1. Download JDK(Java Development Kit)
2. Run the Installer
3. Configure Environment Variables
4. Verify Installation

1.Download JDK:

Go to the official oracle website to download the JDK. Choose x64 MSI Installer on the windows tab and click on download link.

****

2. Run the Installer:

Now, go to your downloads folder and run the installer you just downloaded.

3. Configure Environment Variables:

After installation, you will need to tell your system where to find Java. This is done by setting environment variables.

Locate JDK Path:

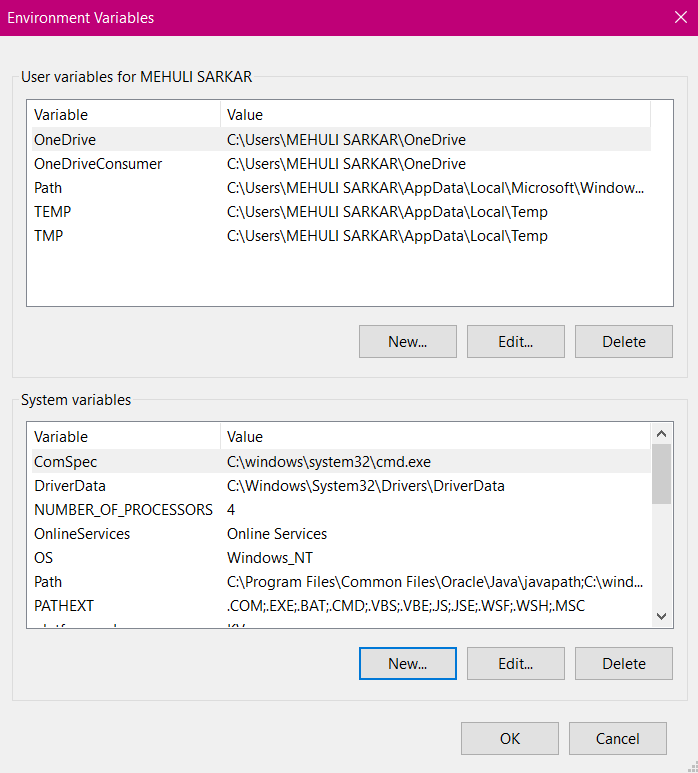
Navigate through your file explorer to reach the JDK installation directory. Normally, it is located at -

C:\Program Files\Java\jdk-22\bin

Copy this path.

Access Environment Variables:

Search environment variable on the terminal. In system properties, click on environment variables. You will be prompted to the screen below.



Update the Path Variable:

Find the Path variable in the System variables section and click on edit. Then, click New and paste your JDK bin path.

Finally, click Ok to close each window.

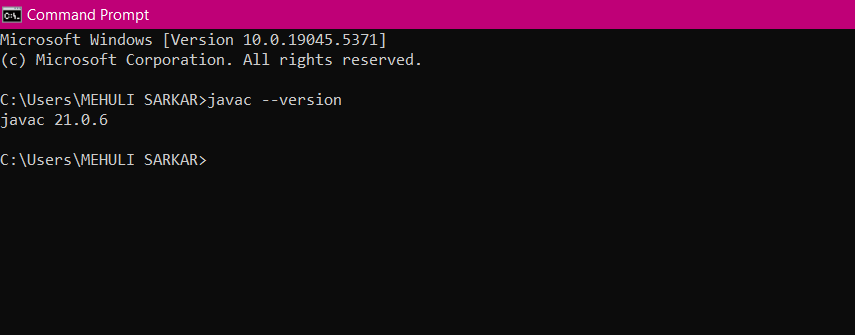
Set JAVA\_HOME Variable:

Back in the environment variables window, under the system variables section, click New to create a new variable.

Now, name the variable  and set its value to the path of the JDK folder directory. Close all the dialogues with the Ok button.

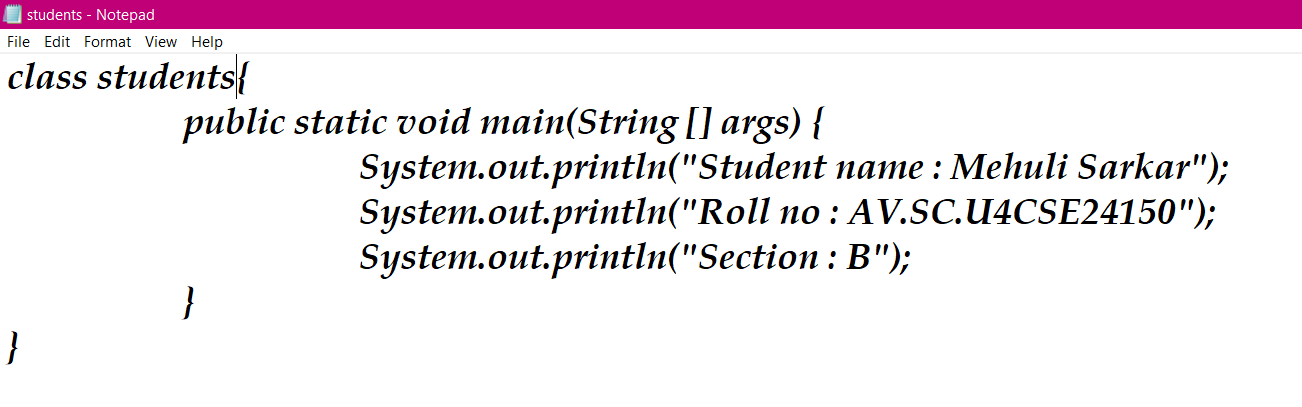
1. Verify Installation:

After the installation, you can verify whether Java is installed by using the following command in the command prompt.

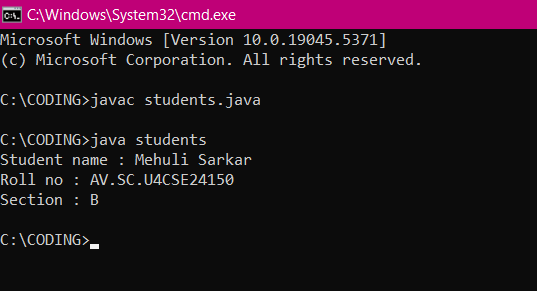


If Java is installed successfully, it will print the version information; otherwise, it will produce an error message indicating that the command is not recognized.

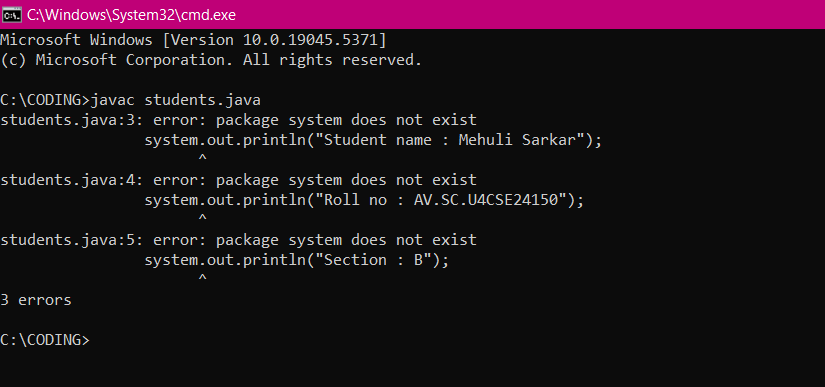
**JAVA Program:**

****

Output:

****

Error:

****:

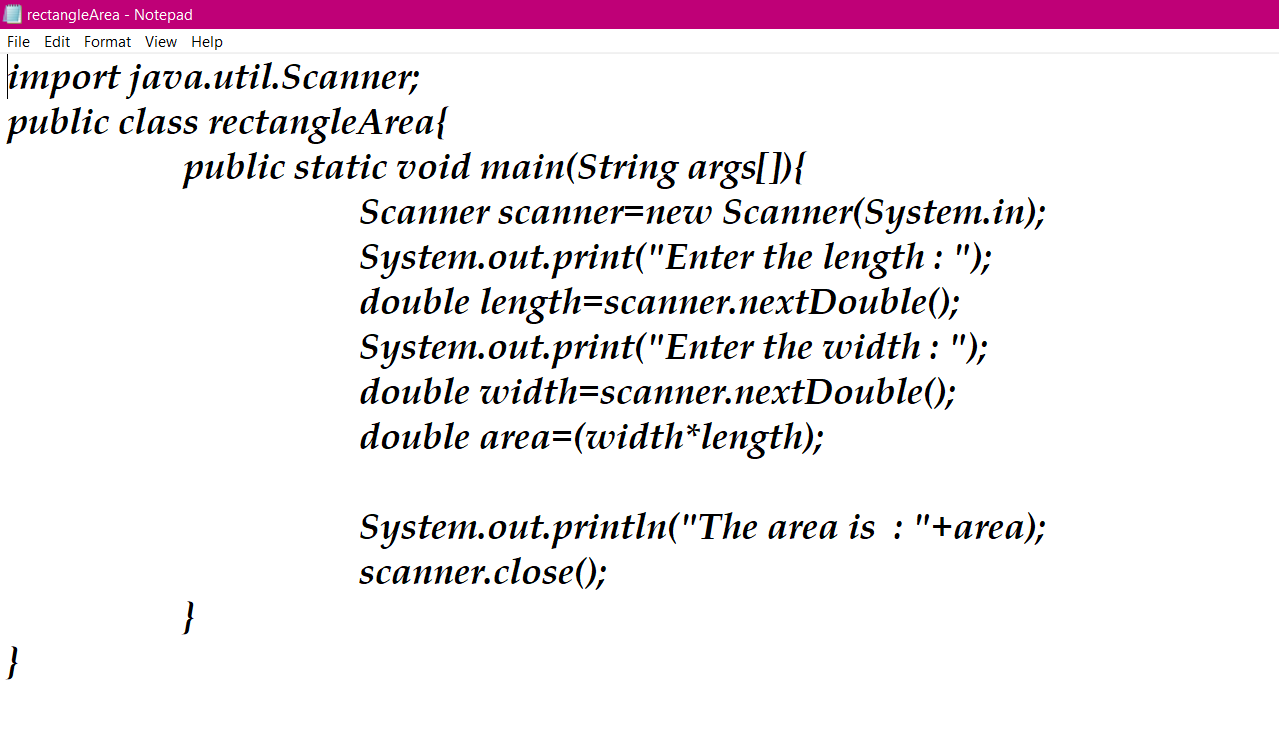
**Week-2**

**1.****Program 1:**

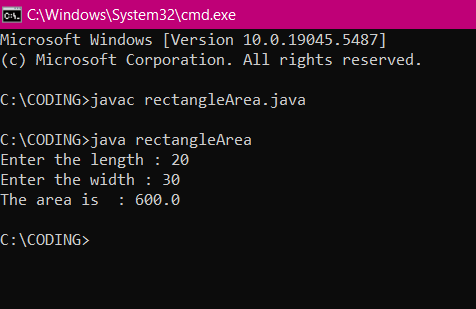
**AIM:**

To find area of rectangle input is given by user

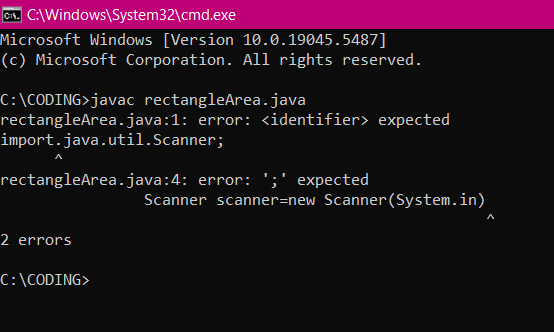
**CODE:**

****

**OUTPUT:**

****

**ERROR TABLE:**

****

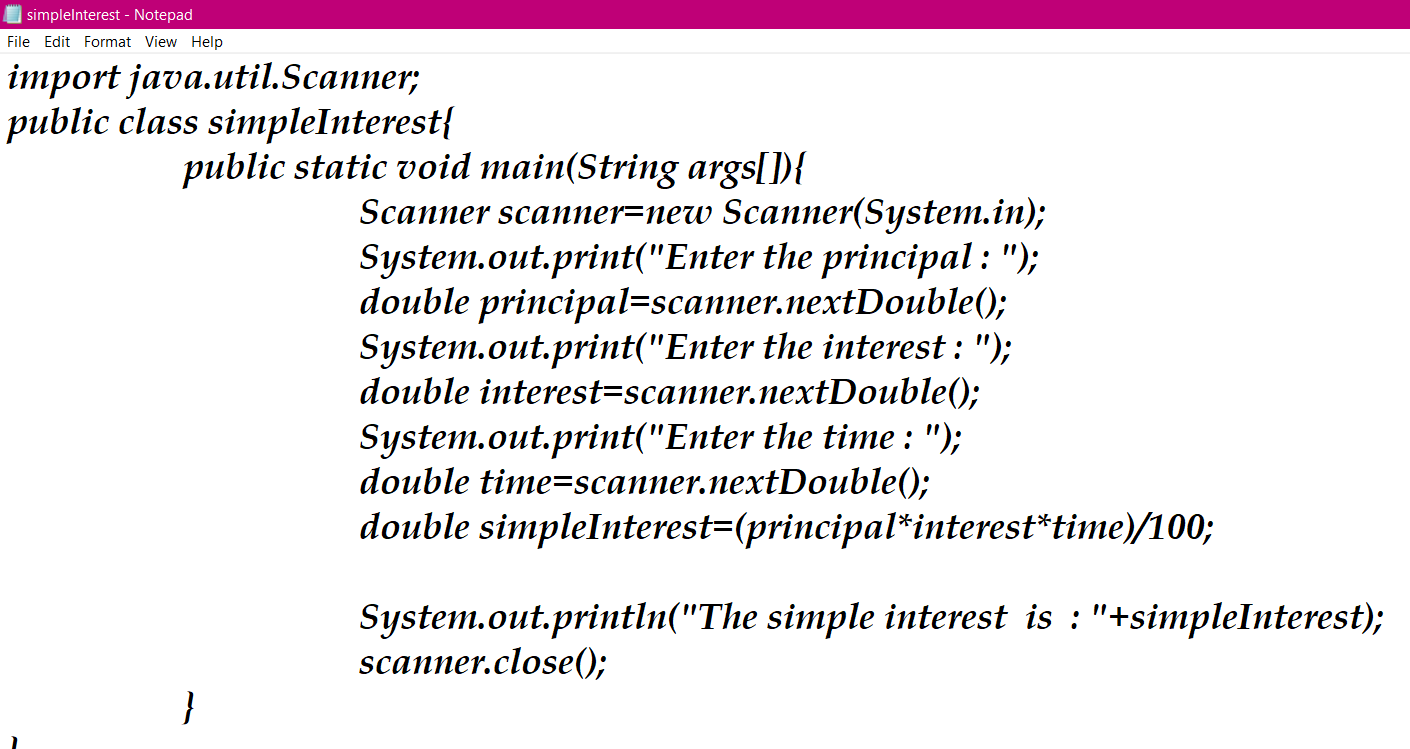
|  |  |
| --- | --- |
| **ERROR** | **RECTIFICATION** |
| Missed semicolon | Added semicolon |

**2. Program 2:**

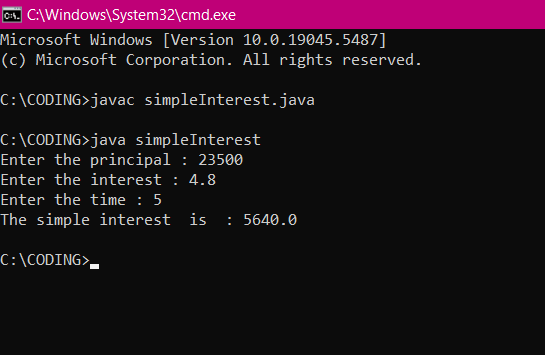
**AIM:**

To find simple interest input given by user

**CODE:**

****

**OUTPUT:**

****

**ERROR TABLE:**

-----

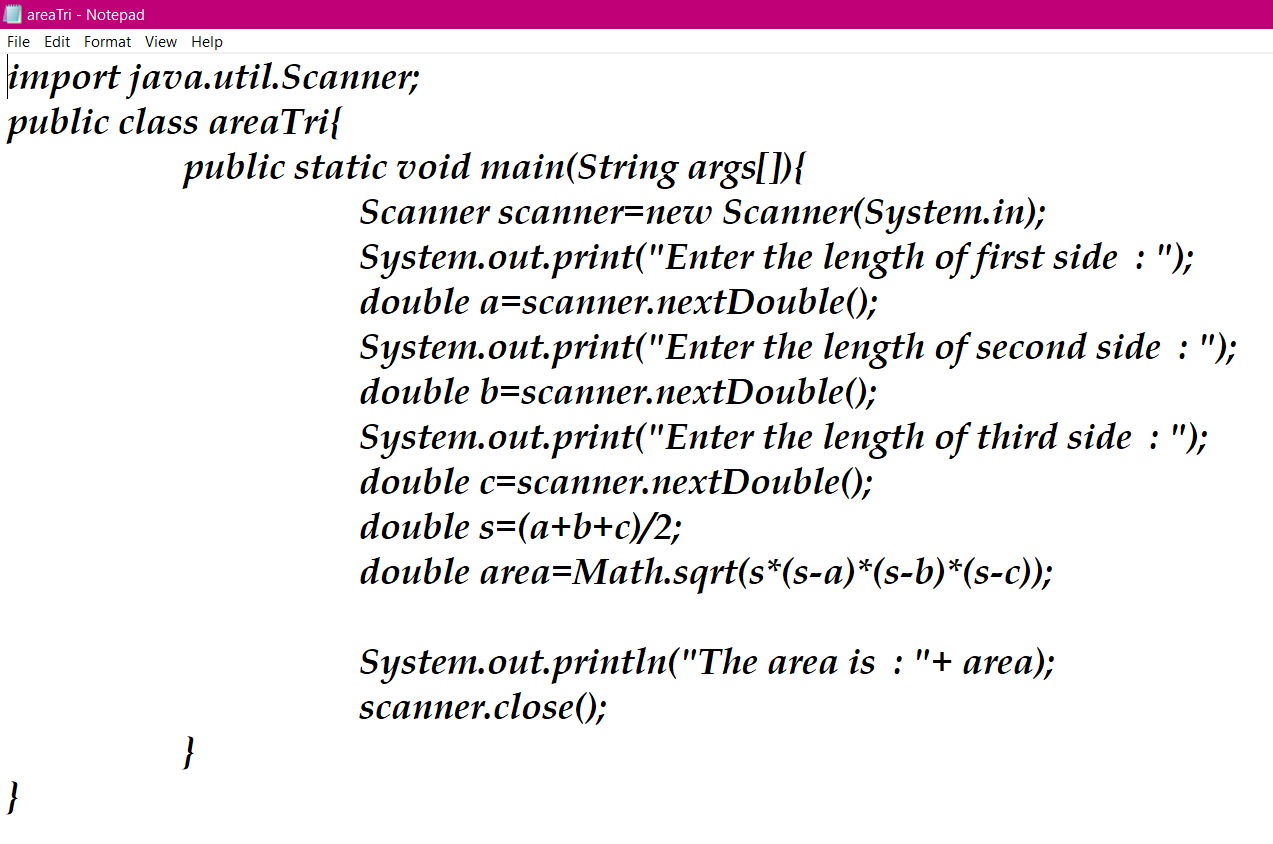
|  |  |
| --- | --- |
| **ERROR** | **RECTICATION** |
| No error | ---- |

**3. Program 3:**

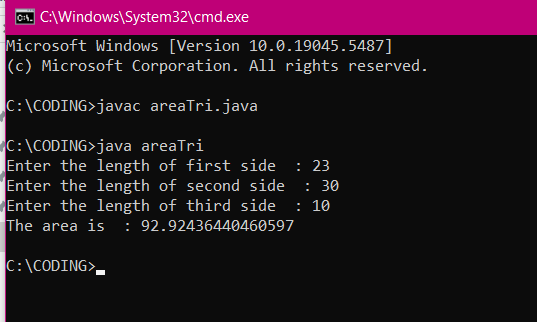
**AIM:**

To find area of triangle input is given by user

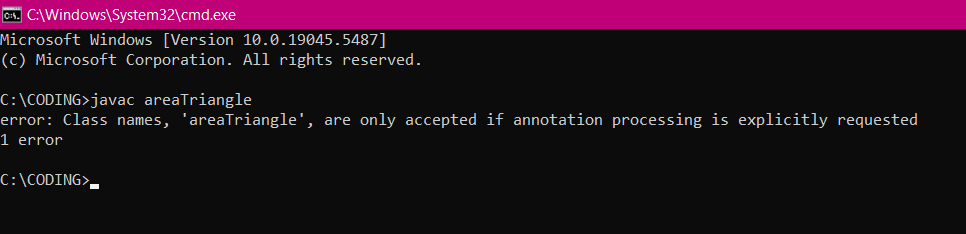
**CODE:**

****

**OUTPUT:**

****

**ERROR TABLE:**

****

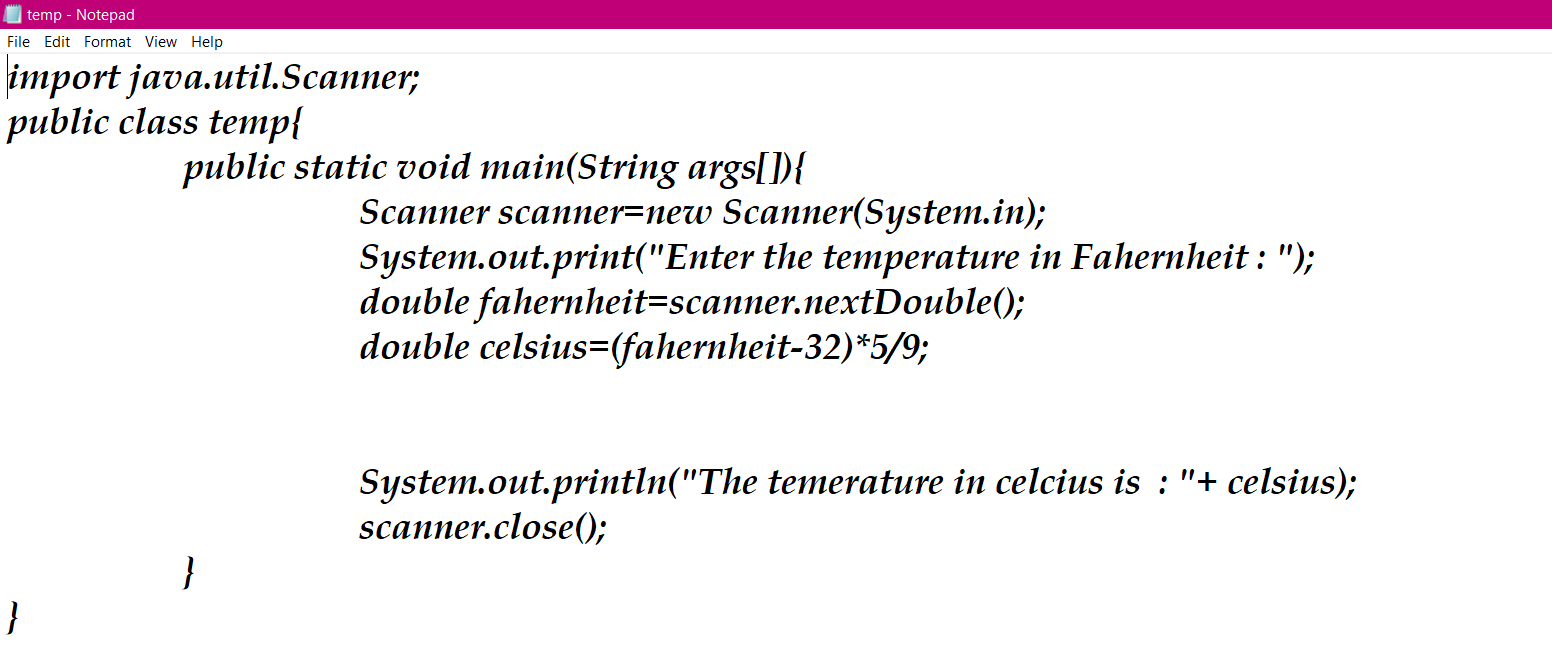
|  |  |
| --- | --- |
| **ERROR** | **RECTIFICATION** |
| Wrong class name | Changed the class name |

**4. Program 4:**

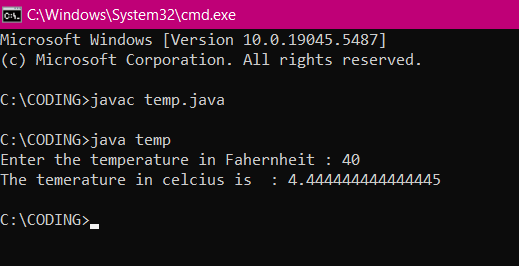
**AIM:**

To convert Fahrenheit to Celsius input is given by user

**CODE:**

****

**OUTPUT:**

****

**ERROR TABLE:**

----

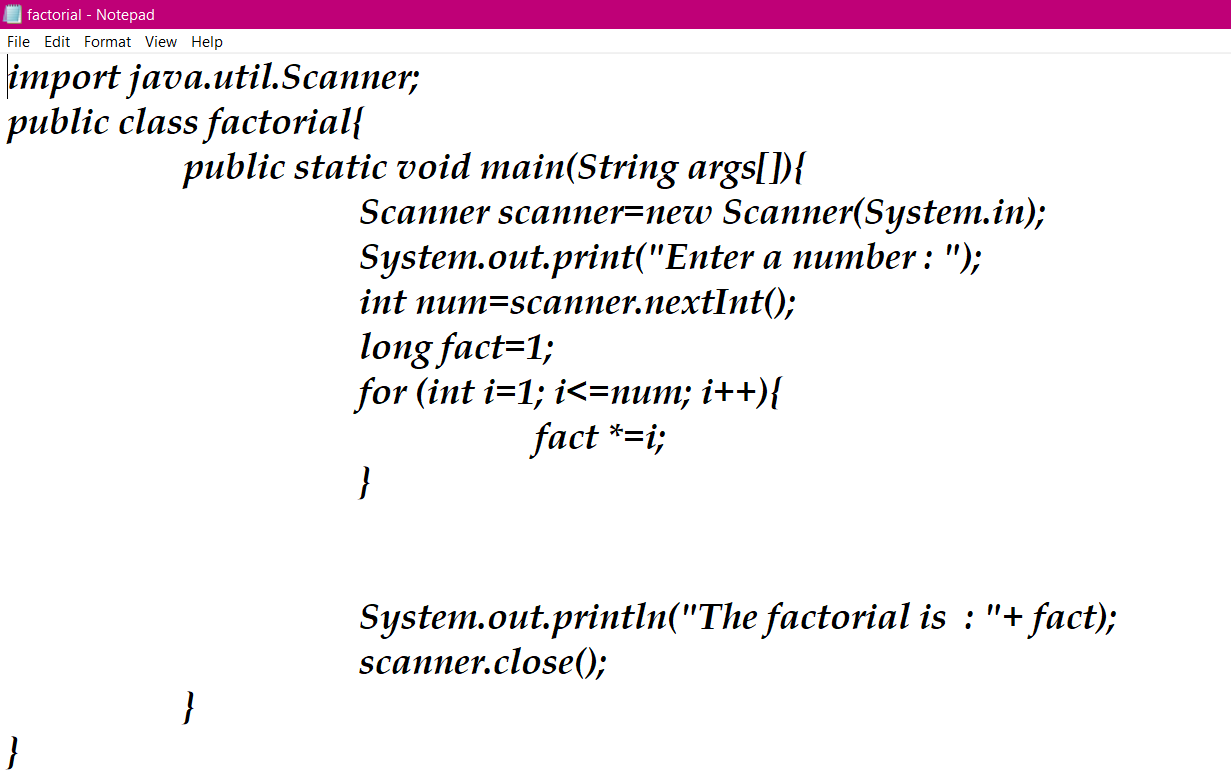
|  |  |
| --- | --- |
| **ERROR** | **RECTIFICATION** |
| No error | ---- |

**5. Program 5:**

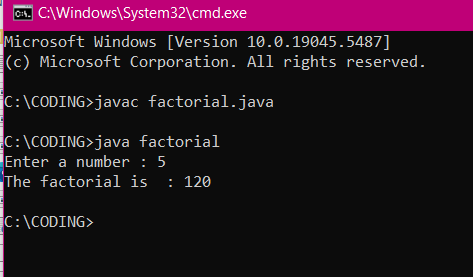
**AIM:**

To find factorial of n

**CODE:**

****

**OUTPUT:**

****

**ERROR TABLE:**

-----

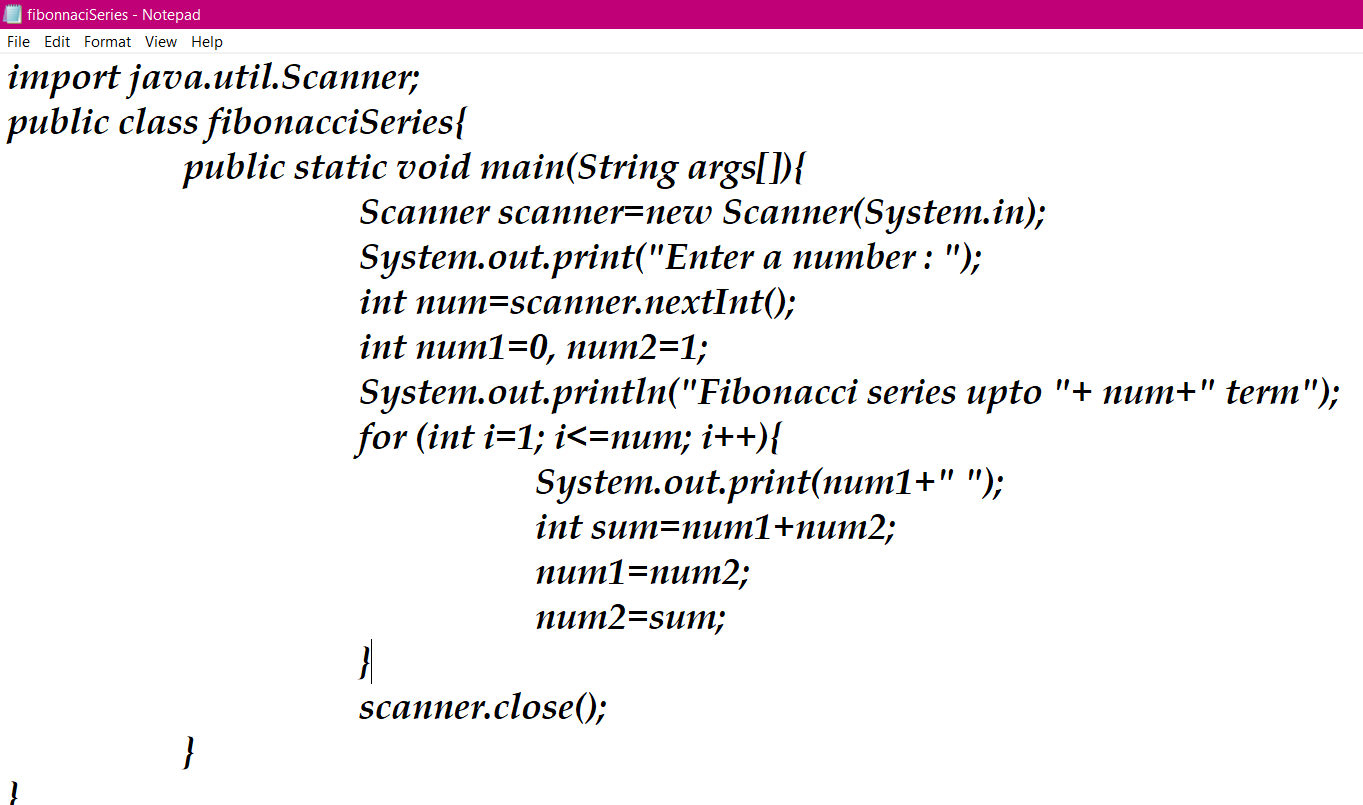
|  |  |
| --- | --- |
| **ERROR** | **RECTIFICATION** |
| No error | ---- |

**6. Program 6:**

**AIM:**

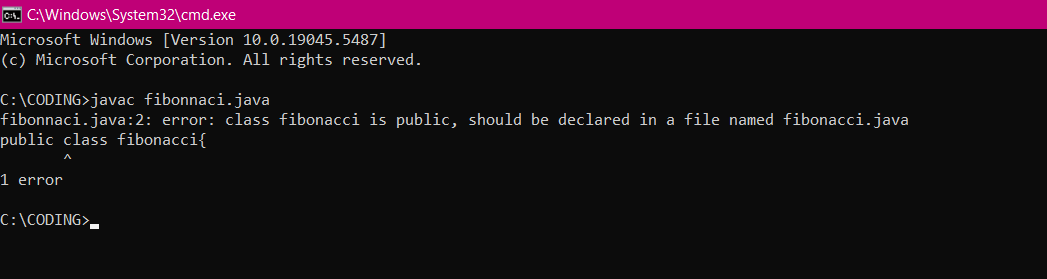
To find Fibonacci sequence

**CODE:**

****

**OUTPUT:**

**ERROR TABLE:**

****

|  |  |
| --- | --- |
| **ERROR** | **RECTIFICATION** |
| Class is public, should be declared in a file named fibonacci.java |  |

**WEEK – 3:**

**PROGRAM- 1:**

**Q. Write a java program through the following instruction:**

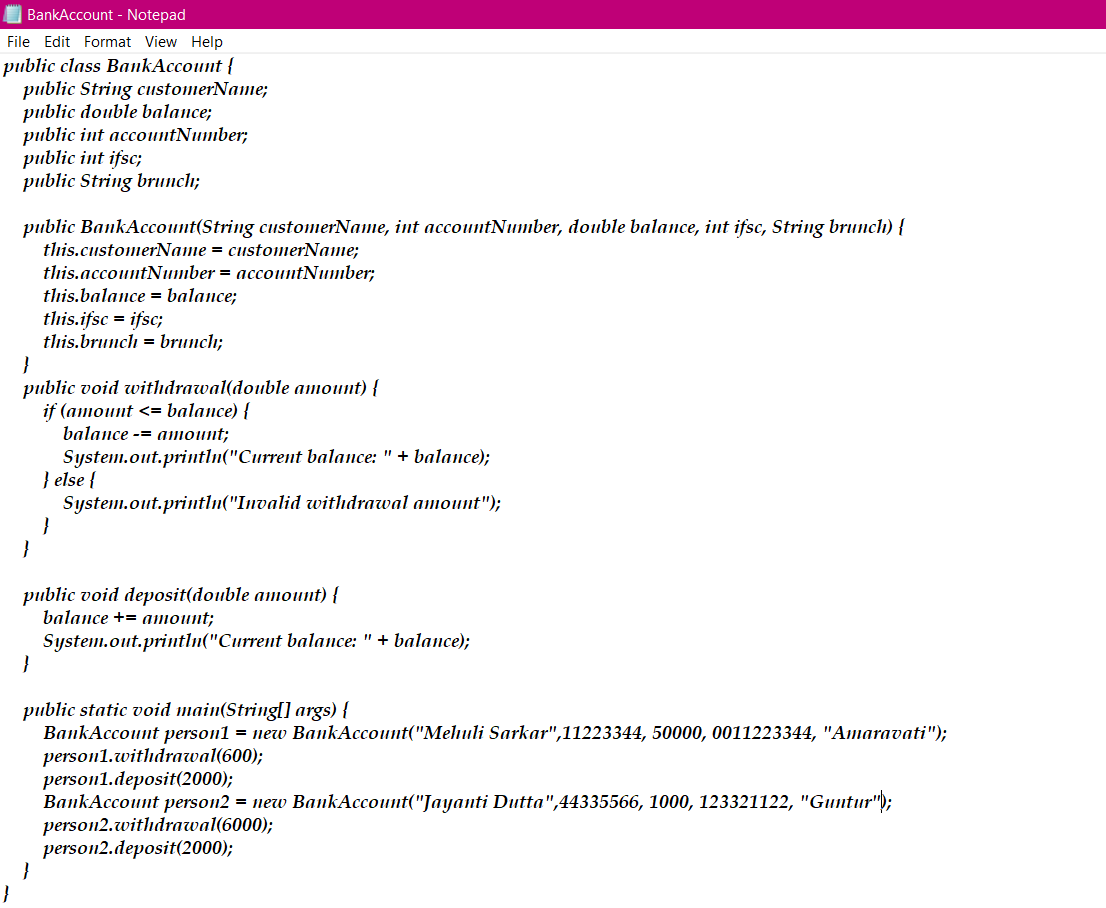
**Create a class named bankAccount with methods deposit and withdraw, where deposit method should accept a parameter and when this method is called the deposited amount should be added to current balance. In addition to that when a withdraw method is called it has to verify if the withdraw amount is less than the current balance, if not then display a message saying insufficient funds.**

**Use the constructor to display the details of the customer.**

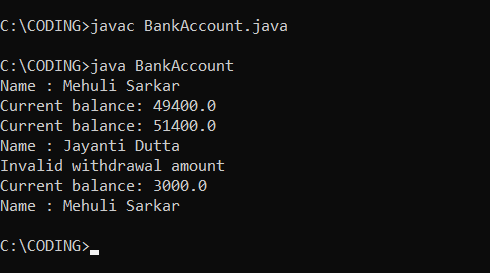
**AIM:**

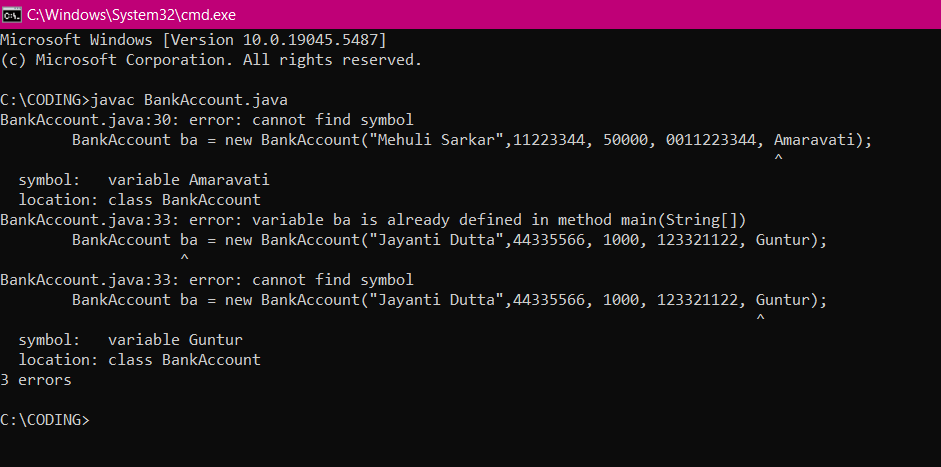
To create a class bankAccount with methods deposit() and withdraw() . create two subclasses savingsaccount and checkingaccount override the withdraw () method in each subclass to impose different withdrawal limits and fees.

**CODE:**



**Output:**



**ERROR:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| 1. Wrong data type of string variable. | 1. Write correct data type. |

**ERROR TABLE:**

**IMPORTANT POINTS:**

1. The condition inside the if statement must be correct.
2. It explains that if the withdrawal money is less than the money in the bank account, then we can withdraw the amount.

**Class diagram:**

|  |
| --- |
| BankAccount |
| - name: String  - Accno: int  - CurrBal: int |
| BankAccount: void  + withdraw(int WAmt): void  + deposit(int DAmt): int |

**PROGRAM- 2:**

**To create java program with following instructions :**

**1. Create a class with name Car**

**2. Create four attributes named car\_color, car\_brand, fuel\_type, mileage**

**3. Create these methods named start(),stop(),service()**

**4. Create the objects named c1, c2,c3.**

**AIM:**

To create java program with following instructions :

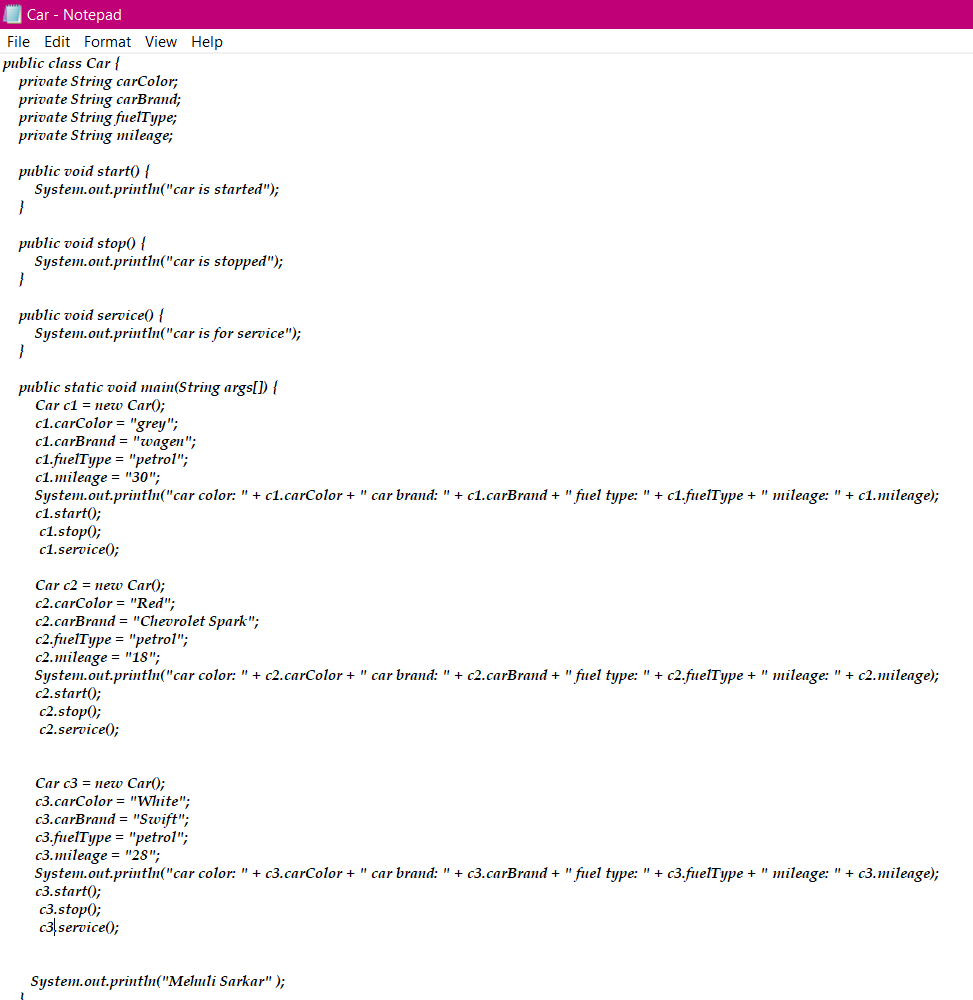
1. Create a class with name Car

2. Create four attributes named car\_color, car\_brand, fuel\_type, mileage

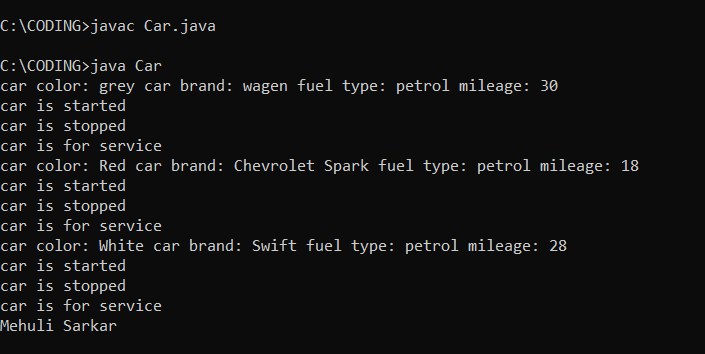
3. Create these methods named start(),stop(),service()

4. Create the objects named c1, c2,c3.

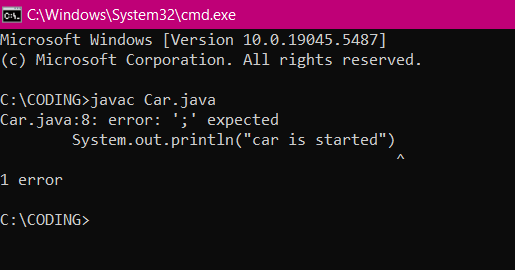
**CODE:**



**Output:**



**ERROR:**

****

**ERROR TABLE:**

|  |  |
| --- | --- |
| **Code Error** | **Code rectification** |
| Not putting the semi-colon | Use semi-colon |

**IMPORTANT POINTS:**

1. Before calling the function we should write the method properly.
2. Here, the “public void start( )” indicates that we are writing a method to call the function.
3. When we call a certain method, the process inside it will be printed as an output of the code.
4. Here the details inside the function are called objects, we can give any objects

**Class diagram:**

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

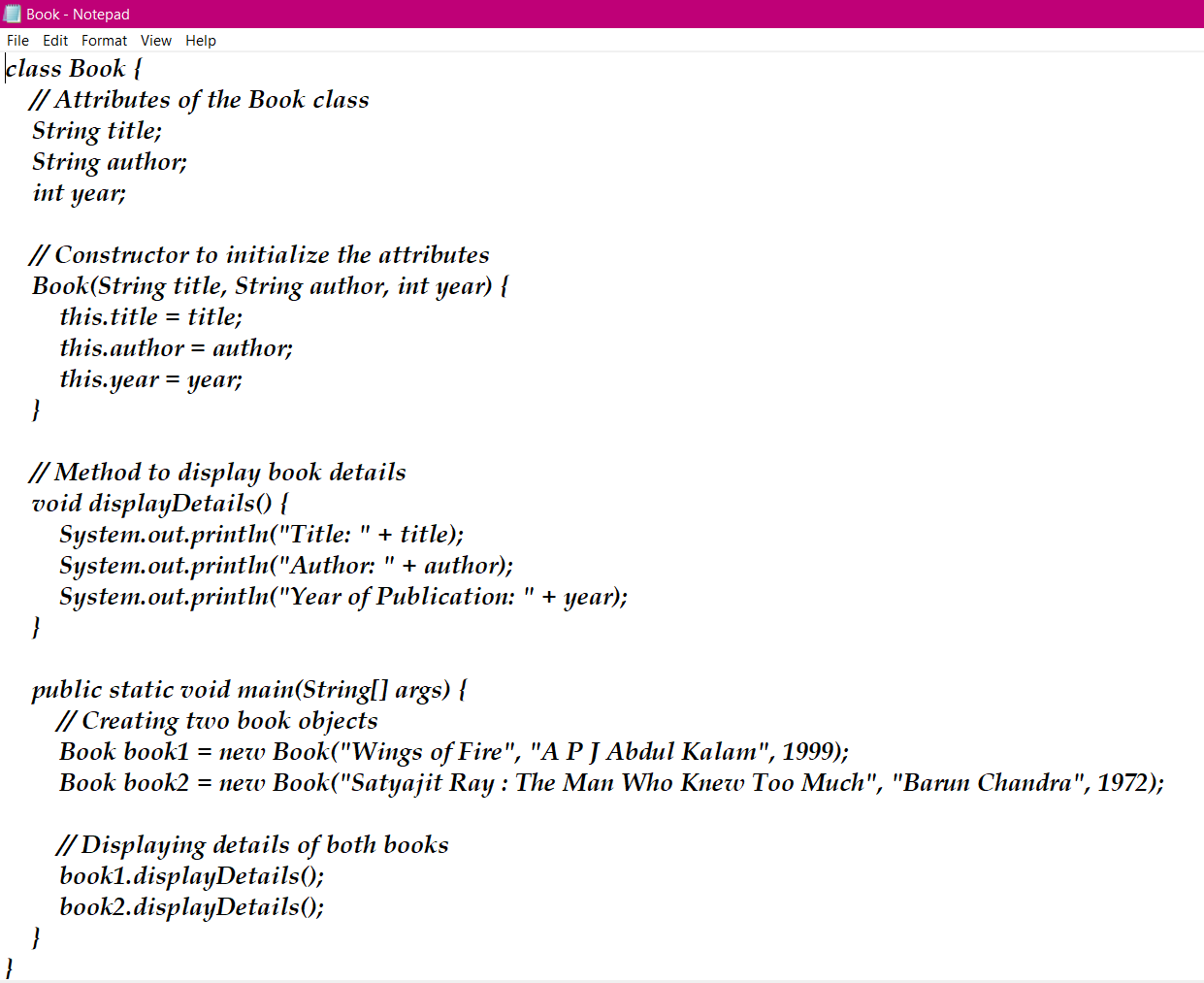
**WEEK 4**

**Program 1**

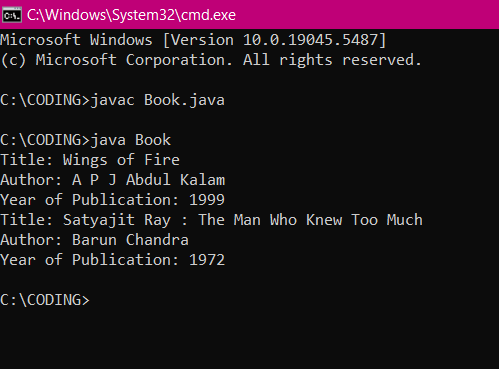
**Aim:**

Write a java program with class named “book” the class should contain various attributes such as the title of the book, author, year of publication. It should also contain a constructor with parameter which initializes the title of the book, author, year of publication. Create a method which displays the details of the book ie title , author, year. Display the details of 2 books by creating 2 objects.

**Code:**



**Output:**

****

**Error:**

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Error message** | **Error rectification** |
| **1.** | **Accessing count in System.out.println without a class reference** | **count is a static variable, meaning it belongs to the class, not an instance. It should be accessed using MyClass.count instead of count inside System.out.println.** |
| **2.** | **Accessing PI through an object** | **Since PI is a final instance variable (not static), it must be accessed through an object, which is correct.** |

**Class Diagram:**

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

**Concepts to be known:**

1)A class is a blueprint, and an object is an instance of that class. (Book is the class, book1 and book2 are objects).

2)Special methods used to initialize objects when they are created (Book(String title, String author, int year)).

3) This keyword refers to the current object’s attributes and prevents naming conflicts.

4) A function inside a class that defines behavior (displayDetails() method prints book details).

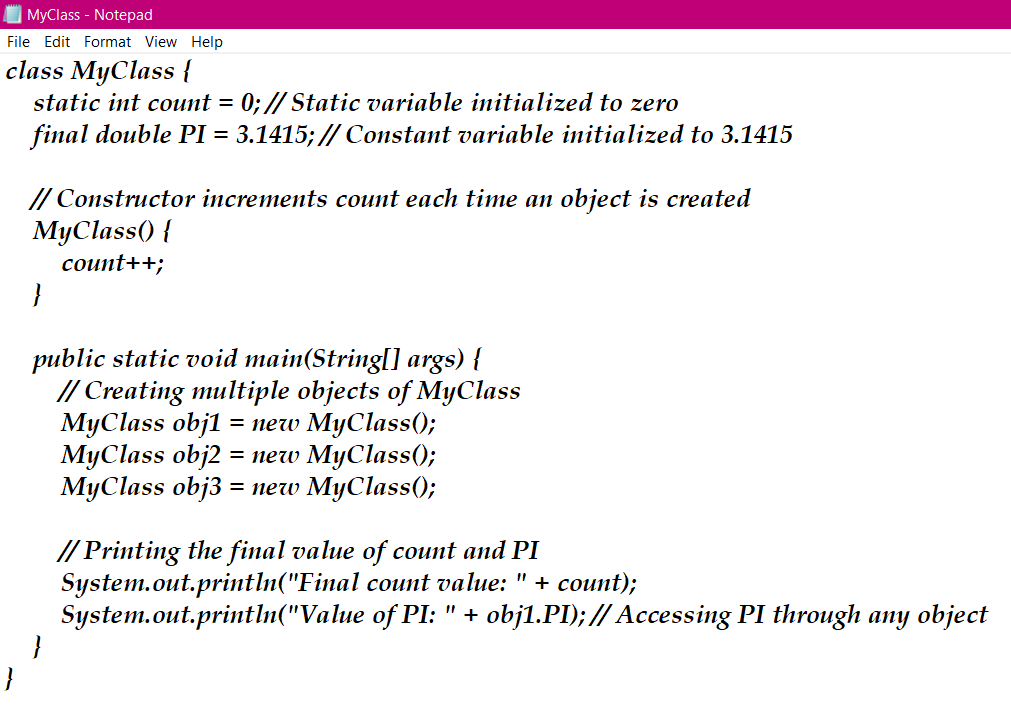
5) Keeping data organized by defining attributes and accessing them through methods instead of direct manipulation.

**Program 2**

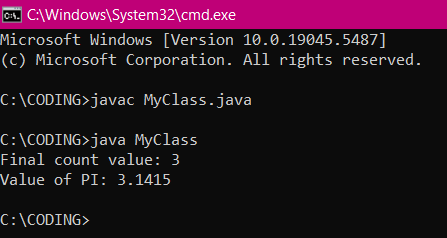
**Aim:**

To create a java program with class named “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. Now define a constructor for my class that increments the count variable each time and object of my class is created. Finally print the final value of “count” and “pi”.

**Code:**

****

**Output:**

****

**Error:**

|  |  |  |
| --- | --- | --- |
| **Sno.** | **Error message** | **Error rectification** |
| **1.** | **Accessing count in System.out.println without a class reference** | **count is a static variable, meaning it belongs to the class, not an instance. It should be accessed using MyClass.count instead of count inside System.out.println.** |
| **2.** | **Accessing PI through an object** | **Since PI is a final instance variable (not static), it must be accessed through an object, which is correct.** |

**Class Diagram:**

|  |
| --- |
| **MyClass** |
| **- static count: int**  **- final PI: double** |
| **+ MyClass()**  **+ main(args: String[]): void** |

**Concepts to be known:**

1. Shared among all instances of a class and accessed using the class name (e.g., MyClass.count).

2) Constants that cannot be modified after initialization (e.g., final double PI = 3.1415).

3) Special methods that execute when an object is created, often used for initializing values or modifying static variables.

4) Instance variables belong to objects, while static variables belong to the class. Static members are accessed using the class name, while instance members require an object.

5) Keeping data safe by defining variables as private (though not used here, it’s a key OOP principle for better control and security).