**Core JAVA Index**

**Date : 05-08-2024**

1. Language and Application
2. Java Features
3. Why Java is Platform Independent
4. OOPS
5. Exceptional Handling
6. Multithreading
7. Web Application
8. Open Source
9. Security
10. Support Networking
11. Memory Management

3. JDK,JRE,JVM

4. Basic Java Programming

1. Variables
2. Methods
3. Conditional Statements
4. Loops

5. Packages

**Date : 06-08-2024**

1. Nested loop : Loop inside loop
2. Arrays
3. Store similar data types
4. Size is fixed
5. Contiguous Memory location
6. Execution is fast

Single Dimensional arrays

Two Dimensional Arrays

3. Logical Programming

4. Switch Case

5. Java.lang

Object class methods

1. equals()
2. hashcode() : Used to know object memory location
3. getclass() : To know about the class information
4. finalize() : Used for garbage collection
5. clone() : Copy of the object
6. wait()
7. notify()
8. notifyall()

6. Scanner Class

It will accept the values from the keyword

7. Enum

Used to store a common information in a place

8. Event Management Application

**Date : 07- 08 - 2024**

* OOPS

Encapsulation

Inheritance

Polymorphism

Abstraction

* Encapsulation

Binding attributes and methods together inside a class and object creation

Programs :

Calculation

Person

Method Flow

* Inheritance :
  + Reusability of existing functionalities from super class to sub class
  + Java won’t support multiple inheritance through classes(extends)
  + Java will support multiple inheritance through classes(implement)
  + Java will support multilevel inheritance through classes
* Polymorphism
  + Compile Time (Method Overloading)
  + Runtime (Overriding)
* Abstraction

Hiding unnecessary details and showing the required information

**Date : 08 - 08 - 2024**

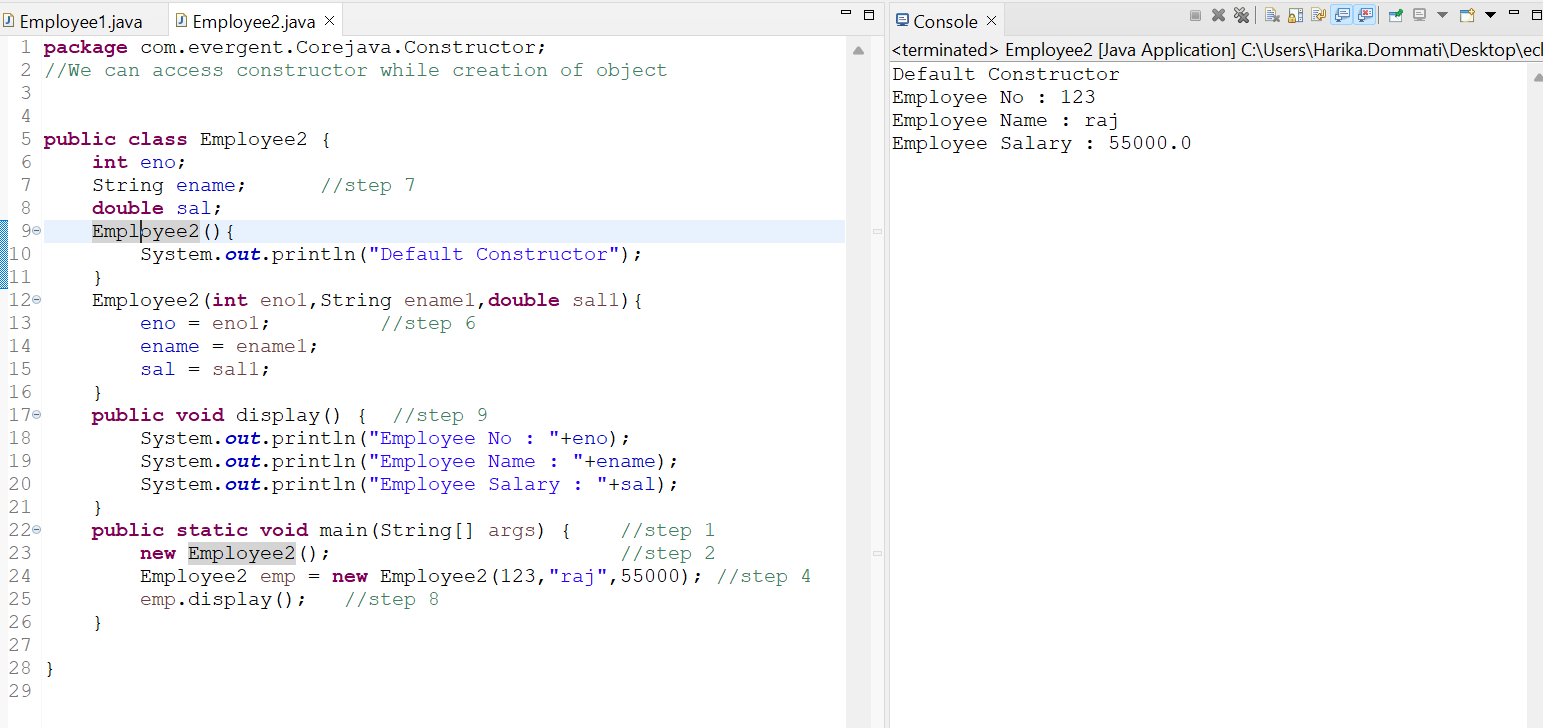
**Constructor**

* Class name and constructor name should be same
* There are 2 types of constructors
  + Default Constructor
  + Parameterized Constructor
* We can access constructor while creation of object
* Constructors are mainly for initializing
* Constructor doesn’t have any return type not even void. If you declare as a void the compiler will consider as a method not a constructor
* Every class needs at least one default constructor
* this, super are keywords
* Always constructor are overloaded
* this keyword always pointing to instance variable
* super keyword always call super class constructor

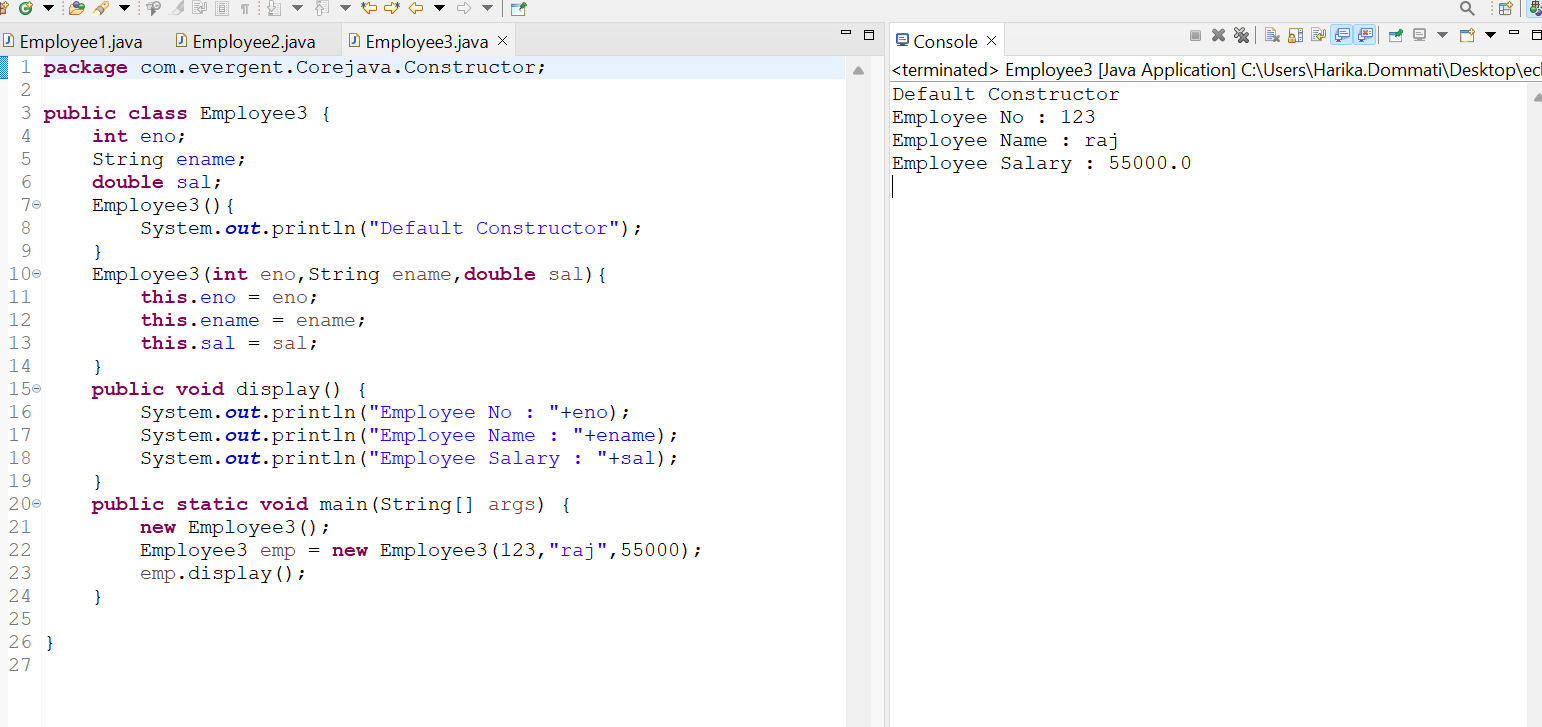
**Program 1 :**

****

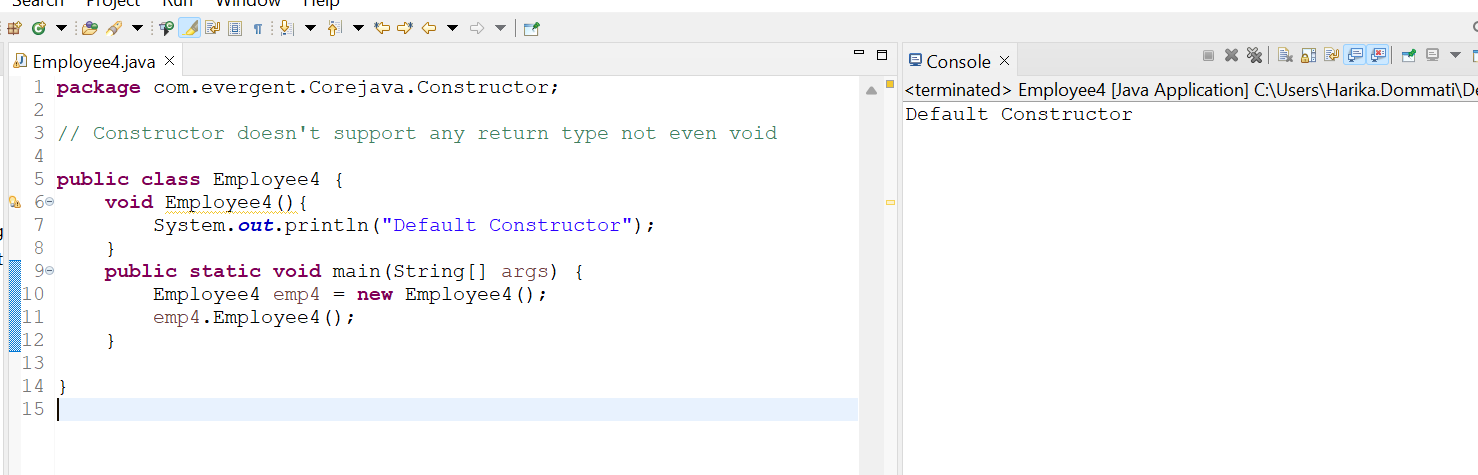
**Program 2 :**

****

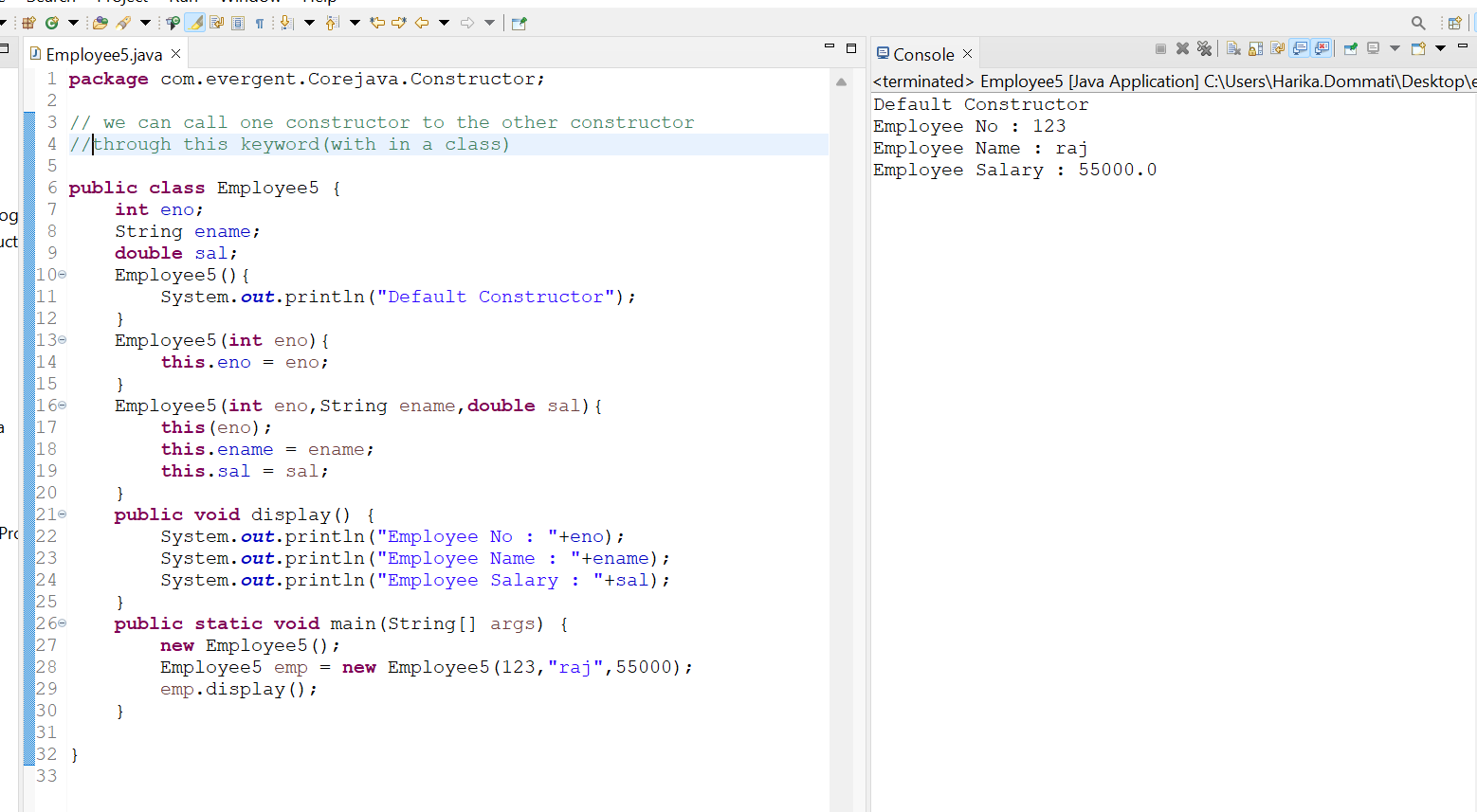
**Program 3 :**

****

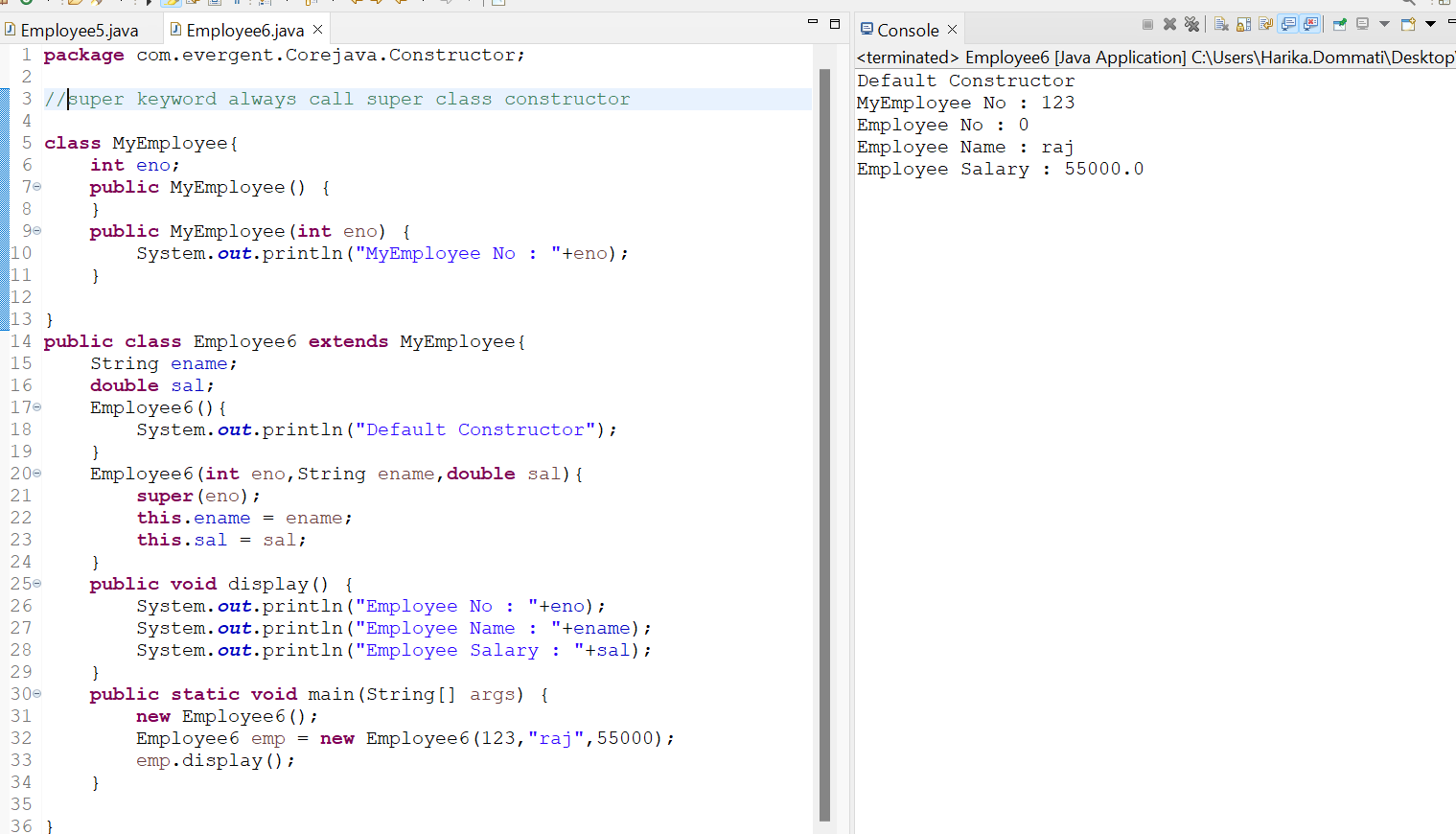
**Program 4 :**

****

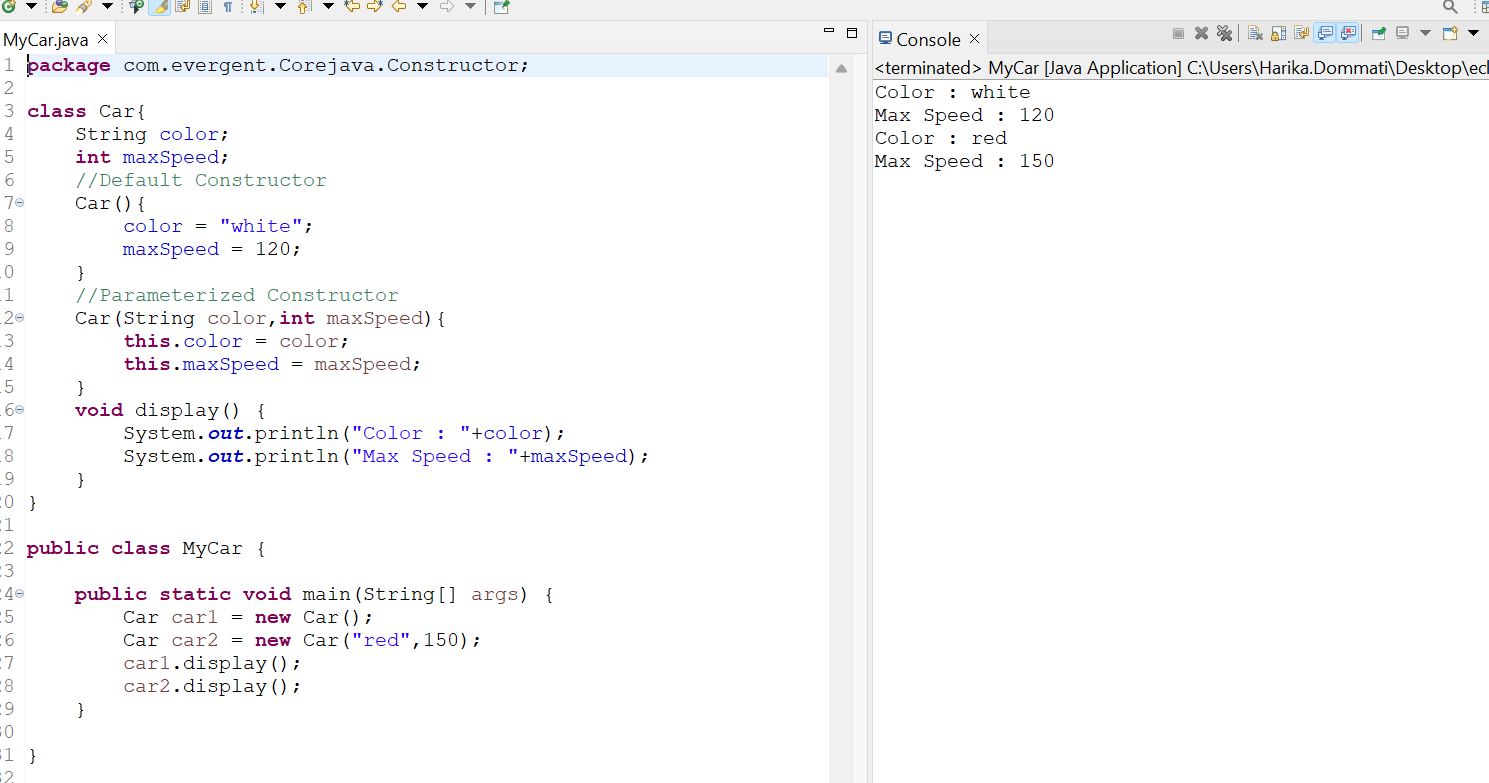
**Program 5 :**

****

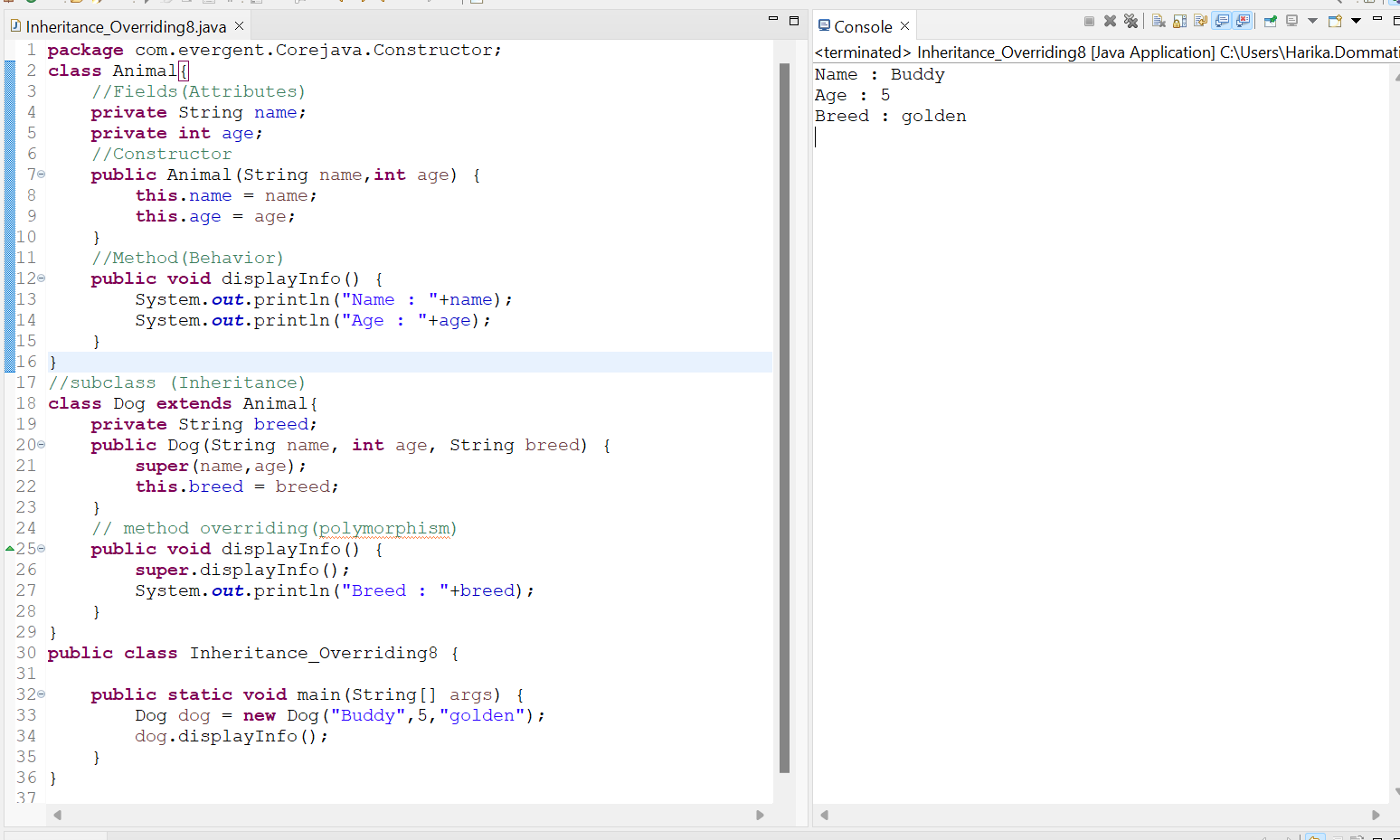
**Program 6 :**

****

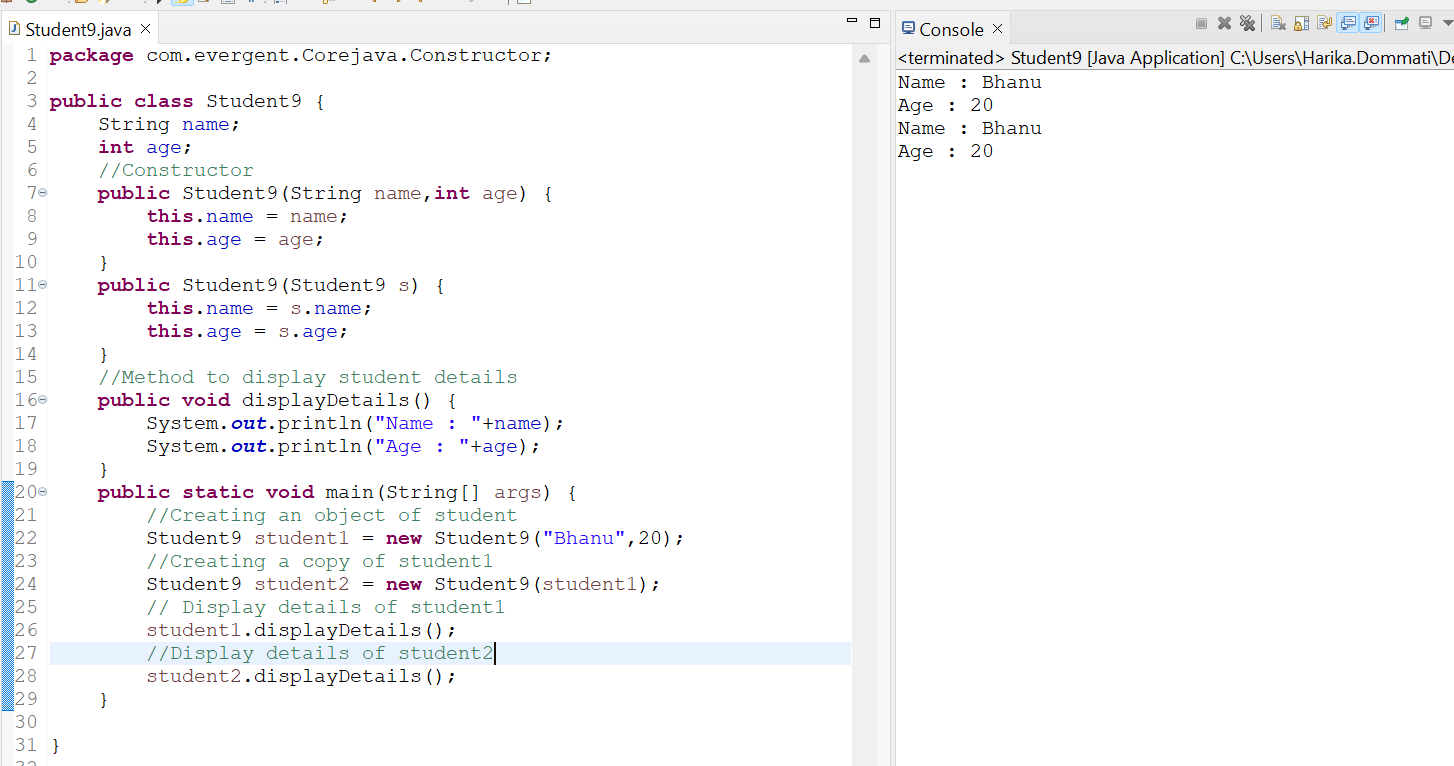
**Program 7 :**

****

**Program 8 :**

****

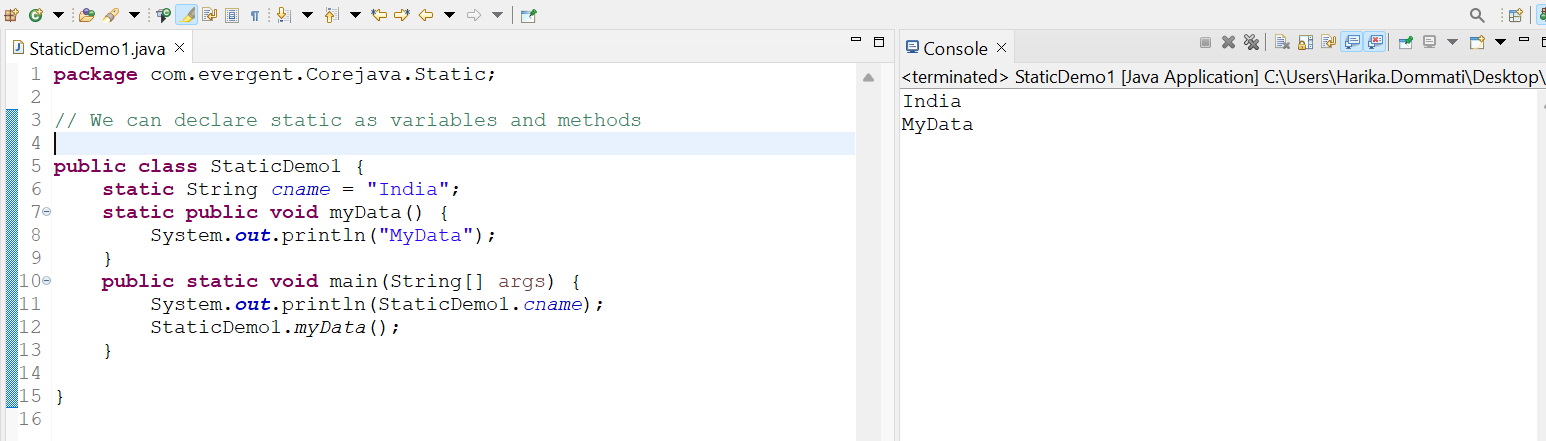
**Program 9 :**

****

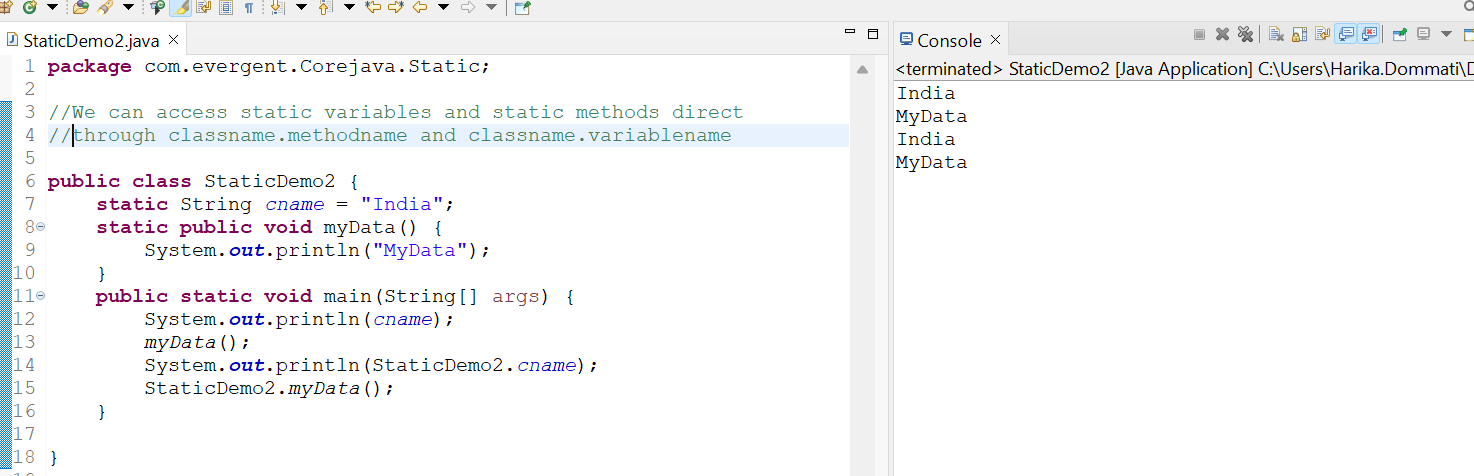
**Static**

* Static is a keyword
* We can declare static as variables and methods
* We can access static variables and static methods direct through classname.methodname and classname.variablename
* Static methods can access only static variables and static methods
* Static methods can not access non static methods and non static variables
* Non static methods can access static methods and static variables
* Static block : Whenever class is loaded in JVM, static blocks are initialized first. Static blocks are used database and networking

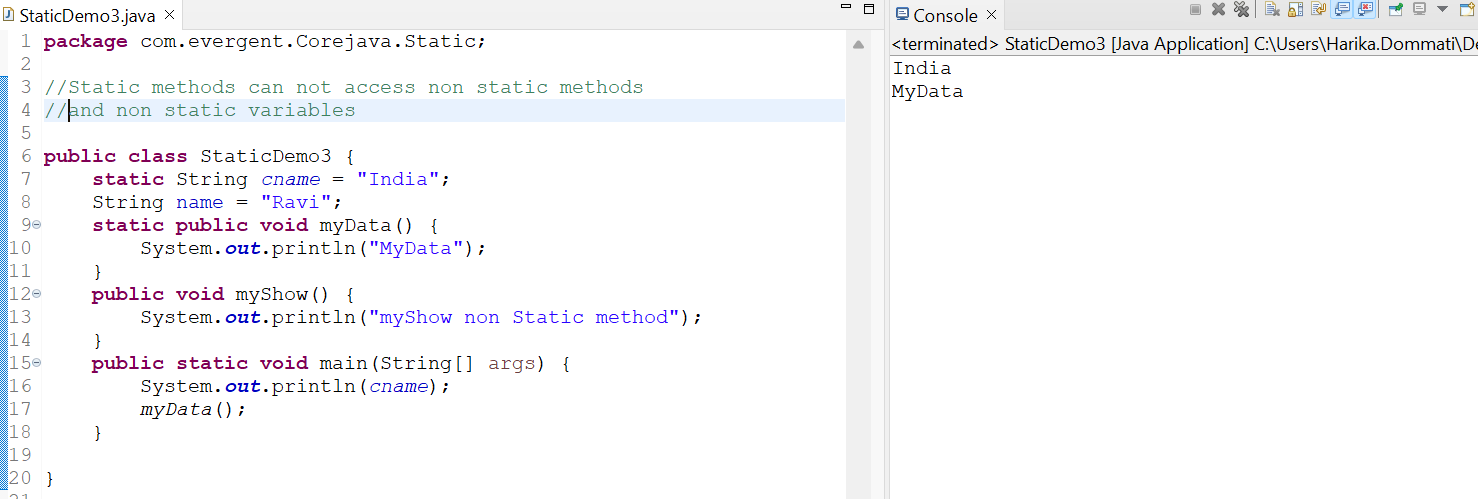
**Program 1 :**

* 

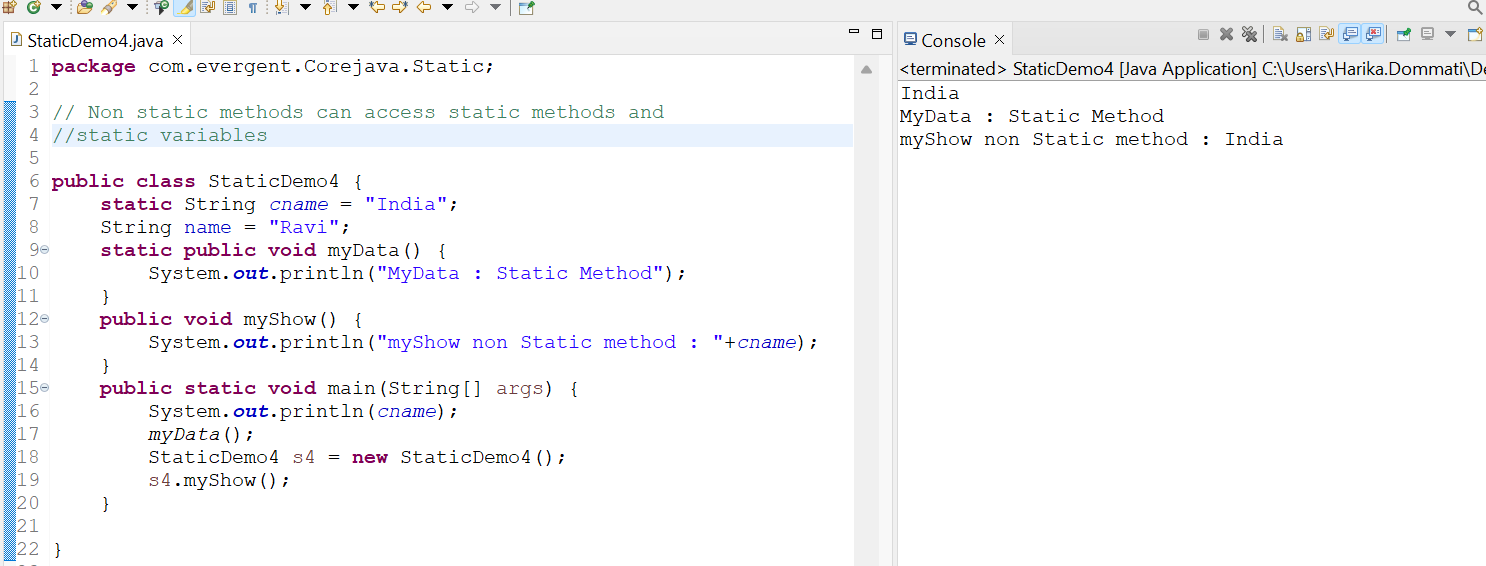
Program 2 :



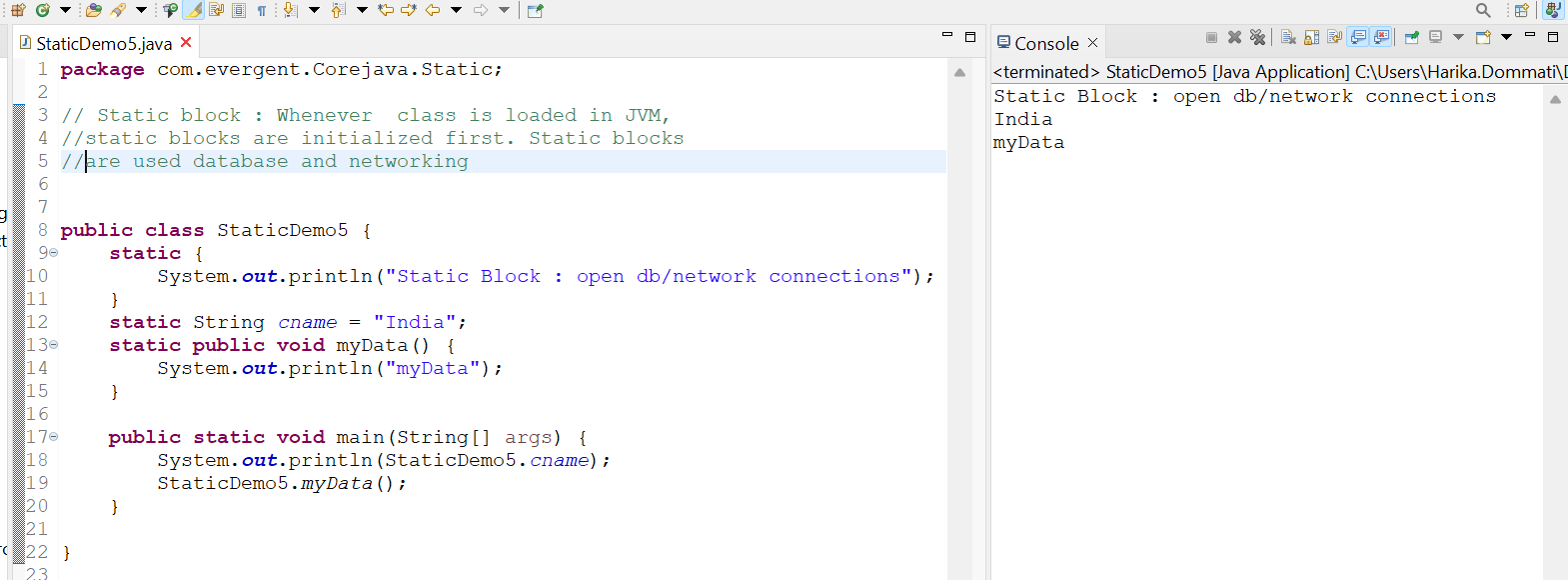
Program 3 :



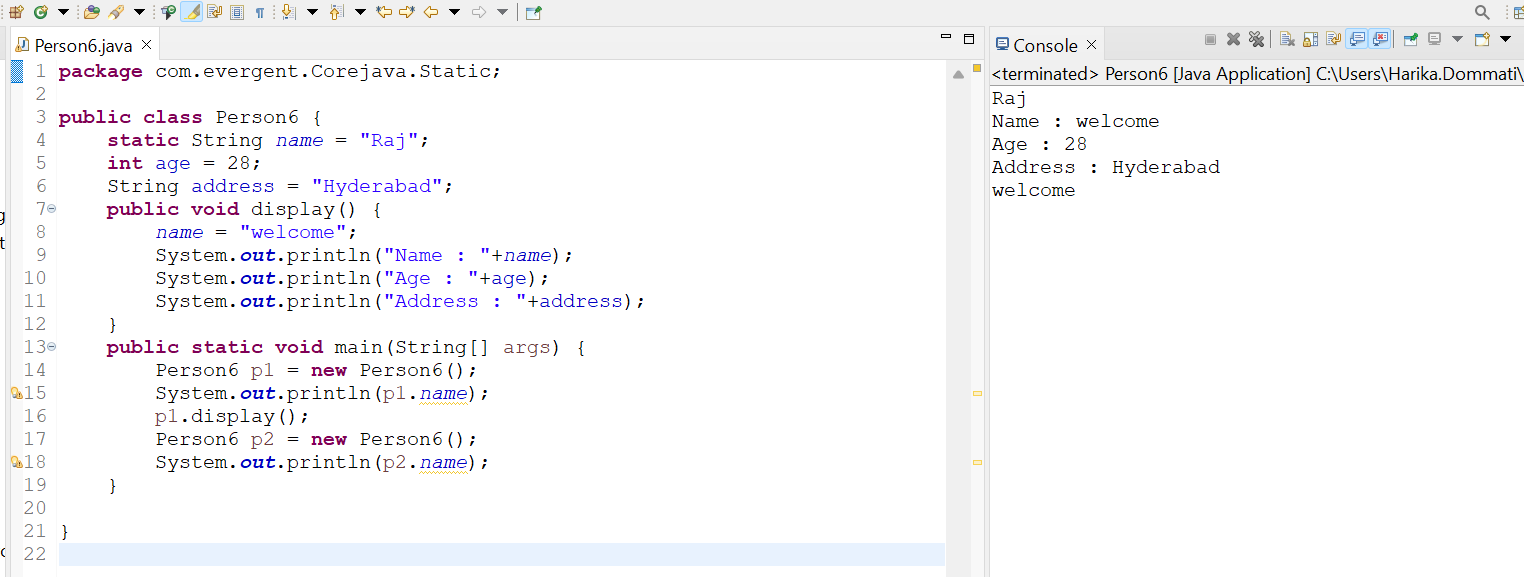
Program 4 :



Program 5 :



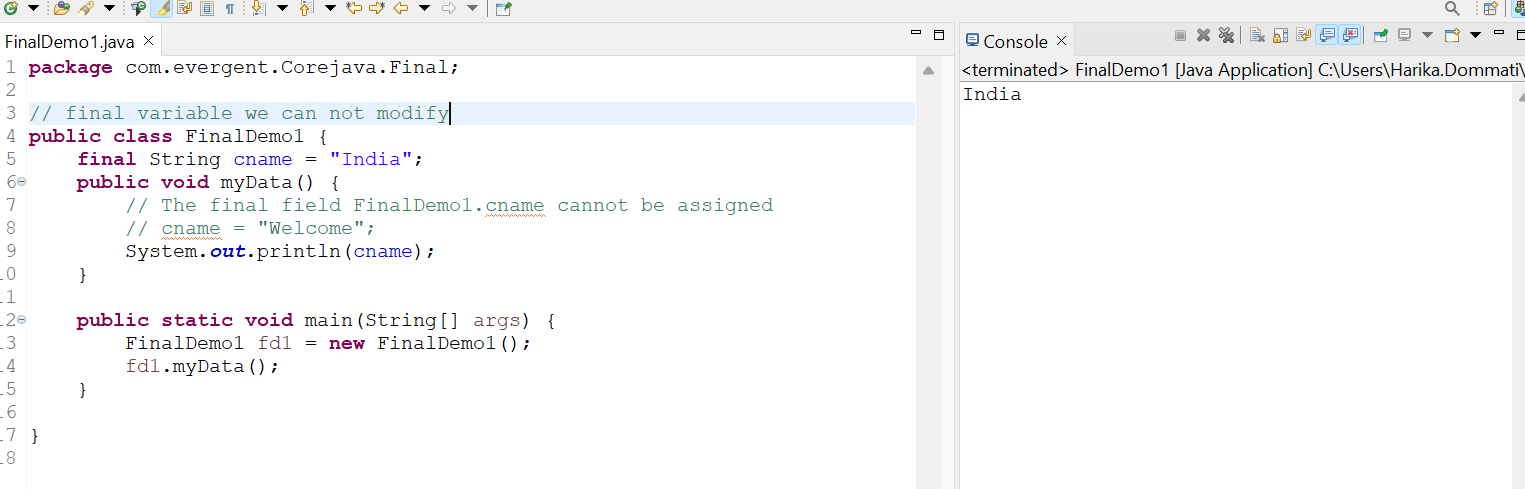
Program 6 :



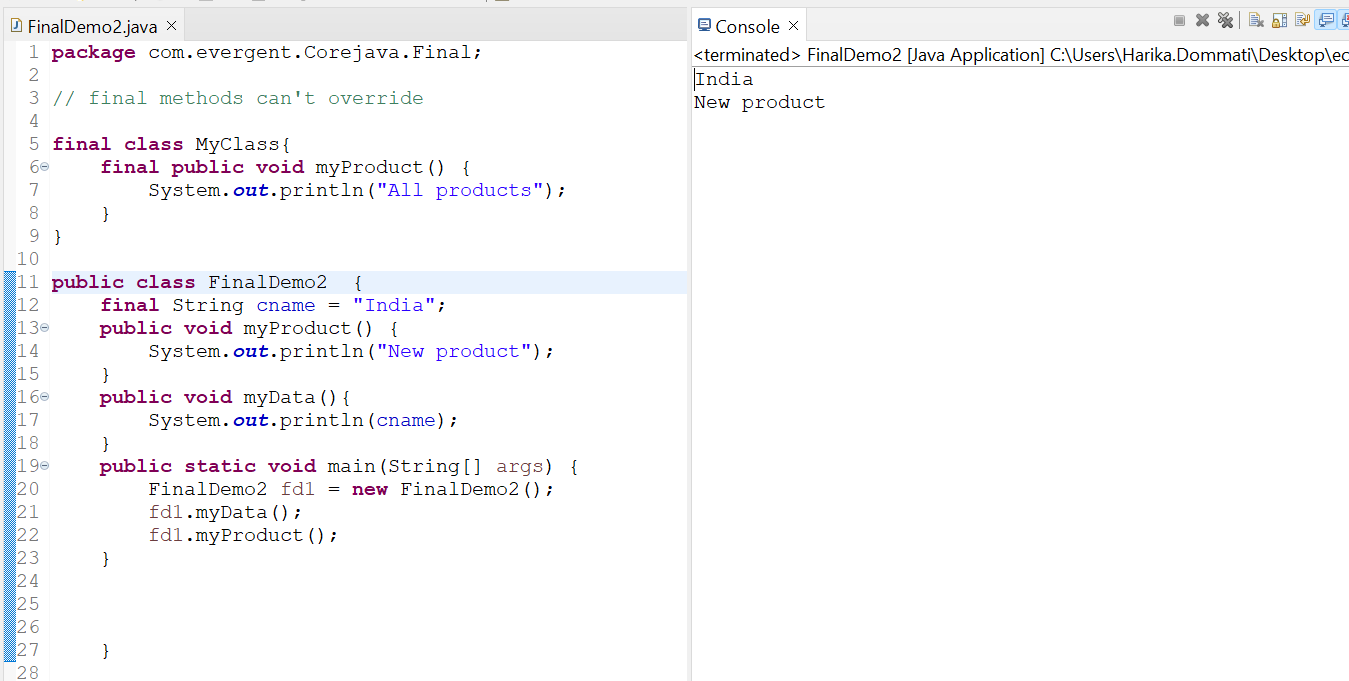
Final

* final is a keyword
* We can declare final as variable, method and class
* final variable we can’t modify
* final methods we can’t override
* final class we can’t extends(Inheritance)
* final variables we can initialize the values while constructor calling

Program 1 :



Program 2 :



Program 3 :



**DAY 6 :**

**Date:12-08-2024**

**Strings** : Strings are there in java.lang package

1.String Class

● String is Final Class

● String class has Methods

● String Methods are Non Synchronized

● String is Immutable Class

● Methods

equals()

toUpperCase()

toLowerCase()

trim()

length()

2.String Buffer

● String Buffer is Final Class

● String Buffer is Mutable

● String Buffer having Methods

● All string methods are Synchronized

● String buffer is not recommended for development but still in API because it’s outdated.

● String Buffer is Thread Safe(Synchronization)

● Methods

append()

insert()

replace()

delete()

reverse()

capacity()

length()

3.String Builder (1.5 version) Tiger Version

● String Builder is Final Class

● String Builder is Mutable Class

● String Builder methods are Non Synchronized

● String Builder having methods

● Methods

append()

insert()

delete()

reverse()

replace()

4.**Strings are Immutable** because when we are trying to create a String Object(new keyword) it is constant.If we are trying to modify an existing String it will create another memory location of the String ,then the Existing object is Eligible for Garbage Collection.

5.In java String is a Sequence of characters used to represent text

6.These are part of java.lang package

7.Immutable strings:Once the object is created it can’t be changed.Any modification string creates a new string object.

8.Strings are stored String pool memory

9.If two strings are created without new and with the same content then they will reference the same object in the string pool.

10.Ways of creating String

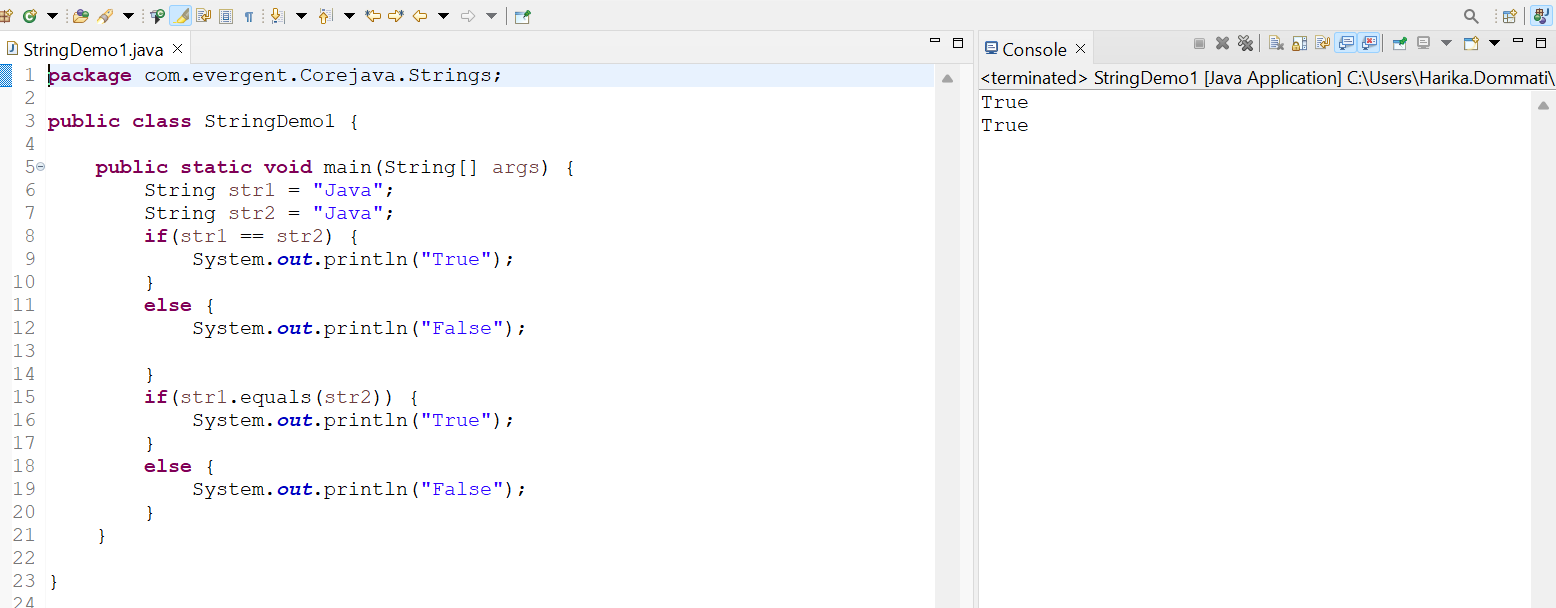
● Using **string literals**

String str=”Hello world”;

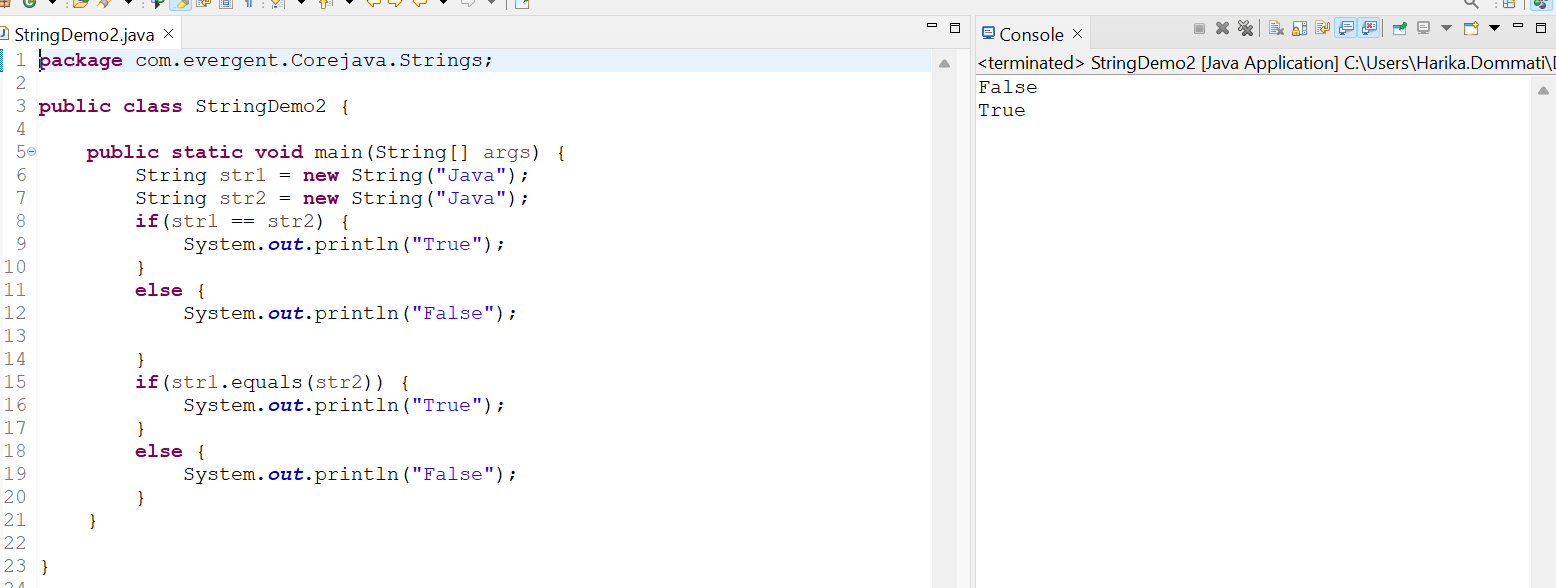
● Using **new** keyword

String str=new String(“Hello World”);

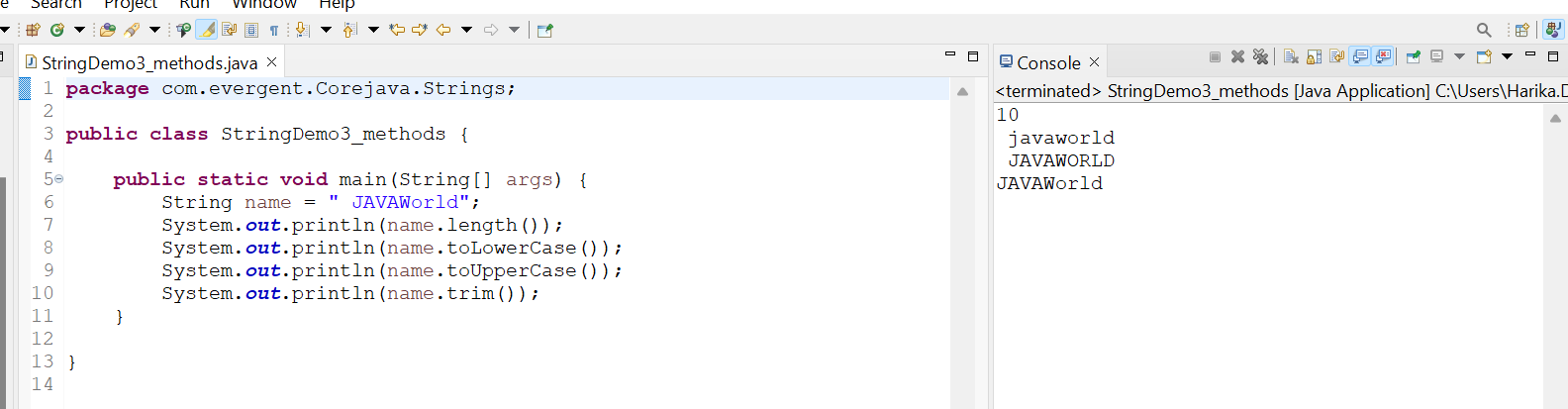
**Program 1 :**

****

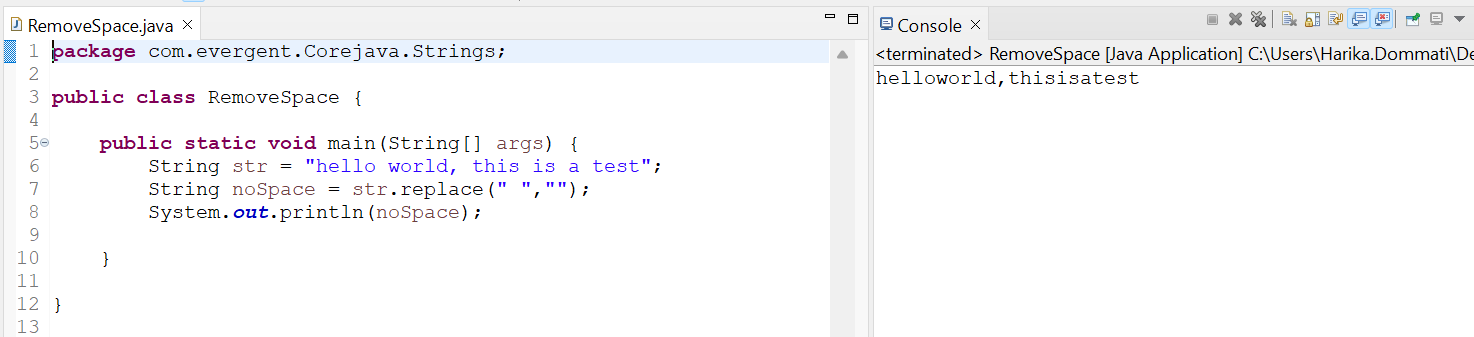
Program 2 :



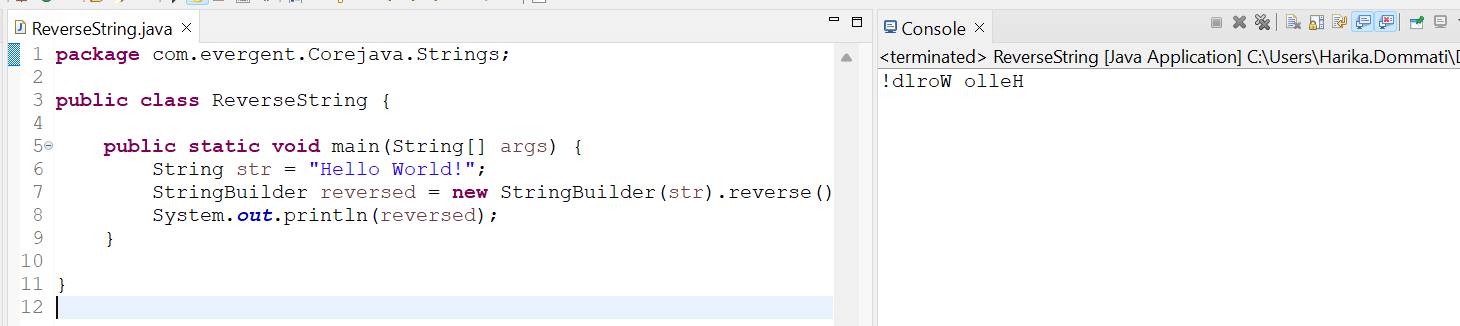
Program 3 :



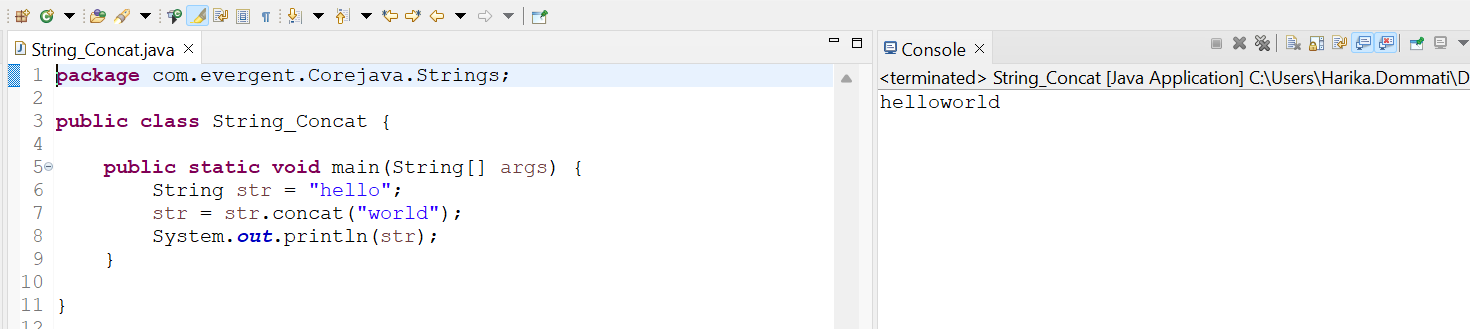
Program 4 :



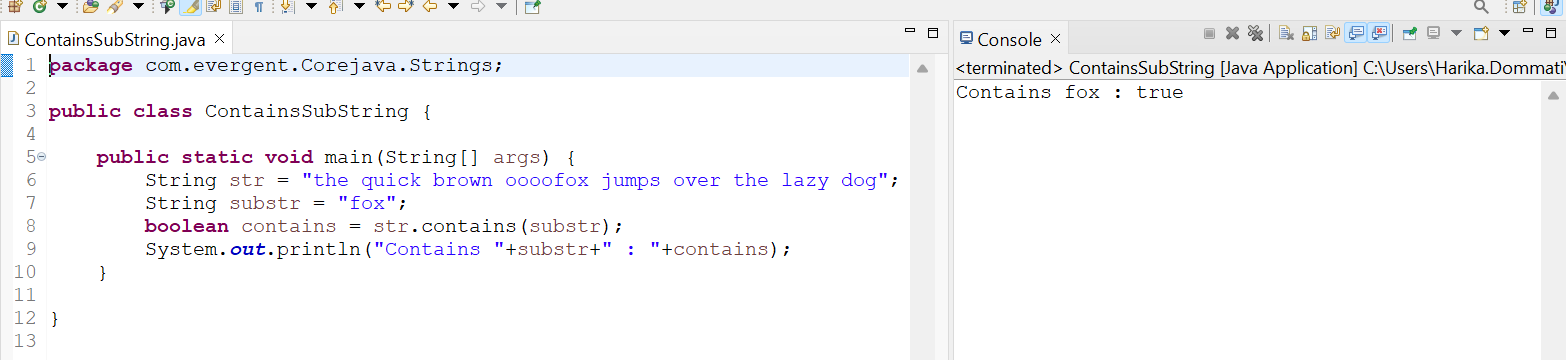
Program 5 :



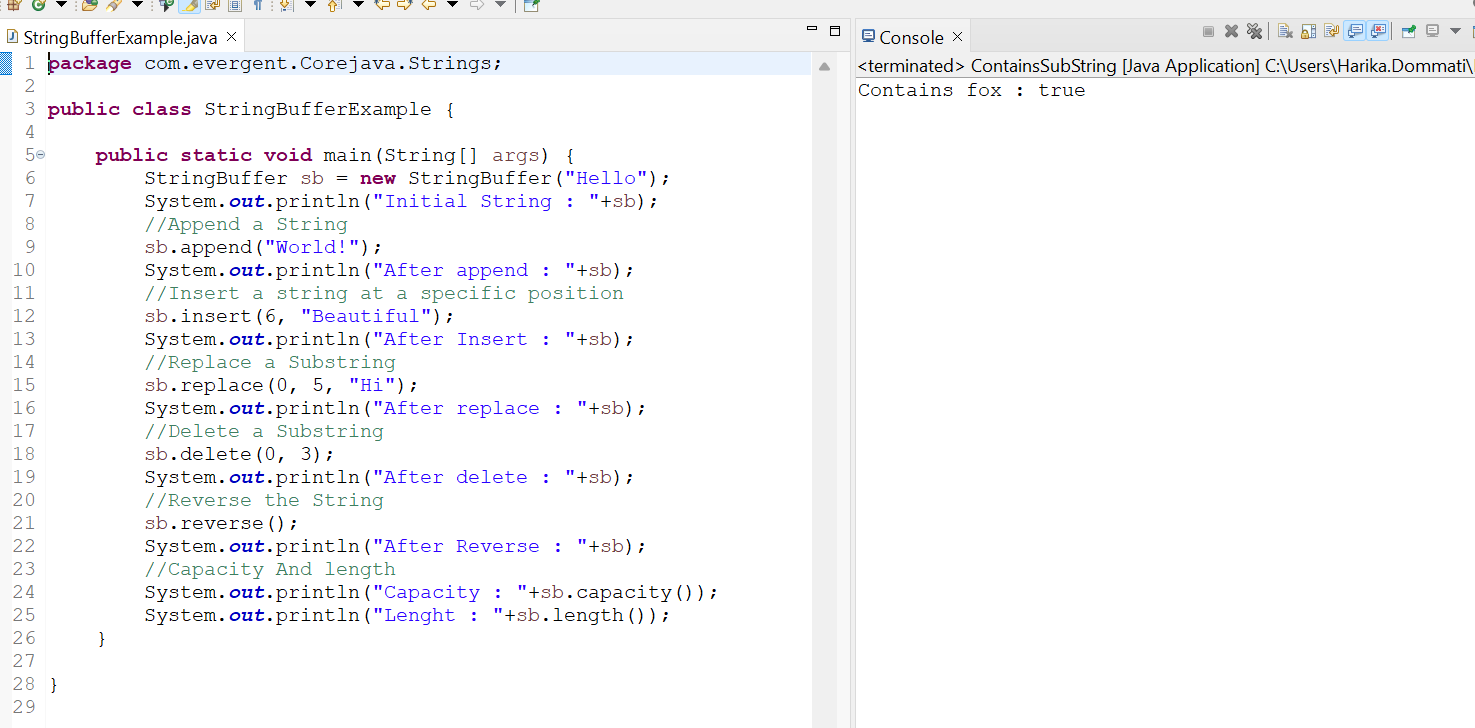
Program 6 :



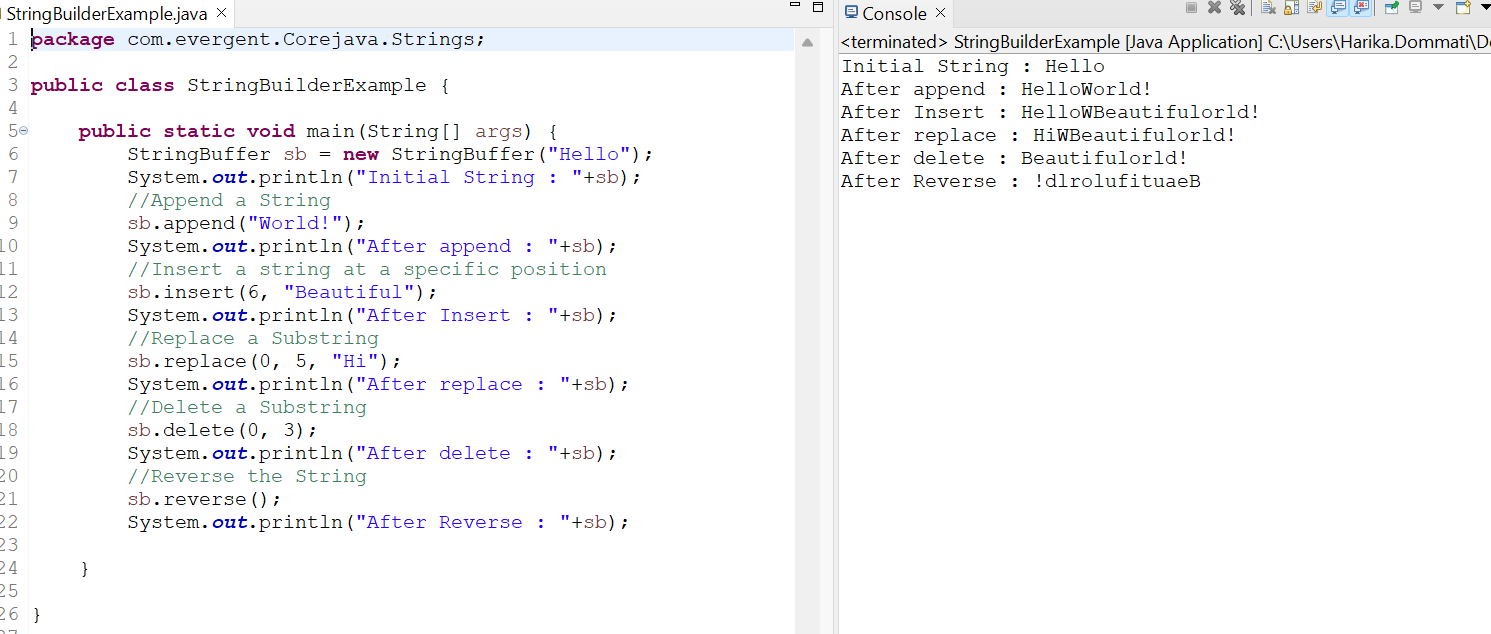
Program 7 :



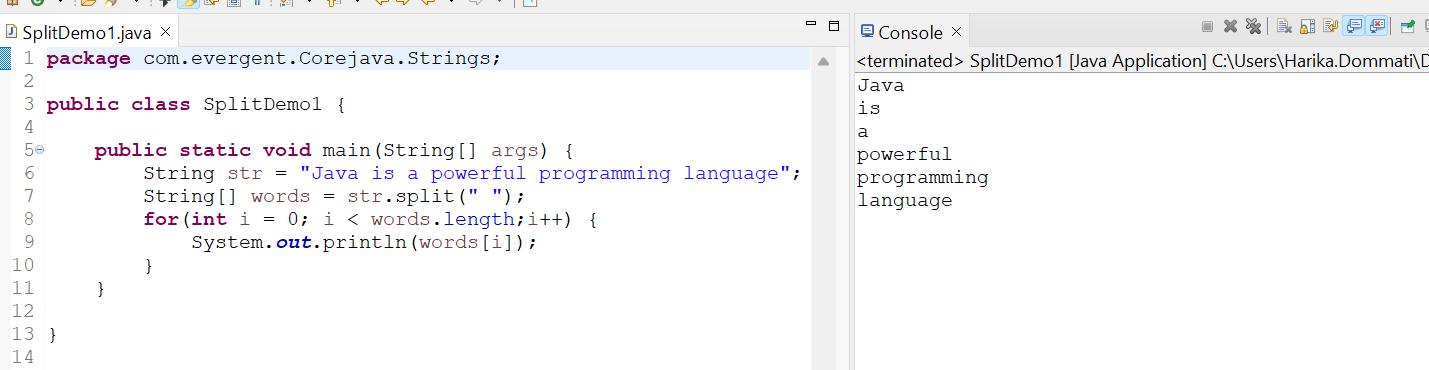
Program 8 :



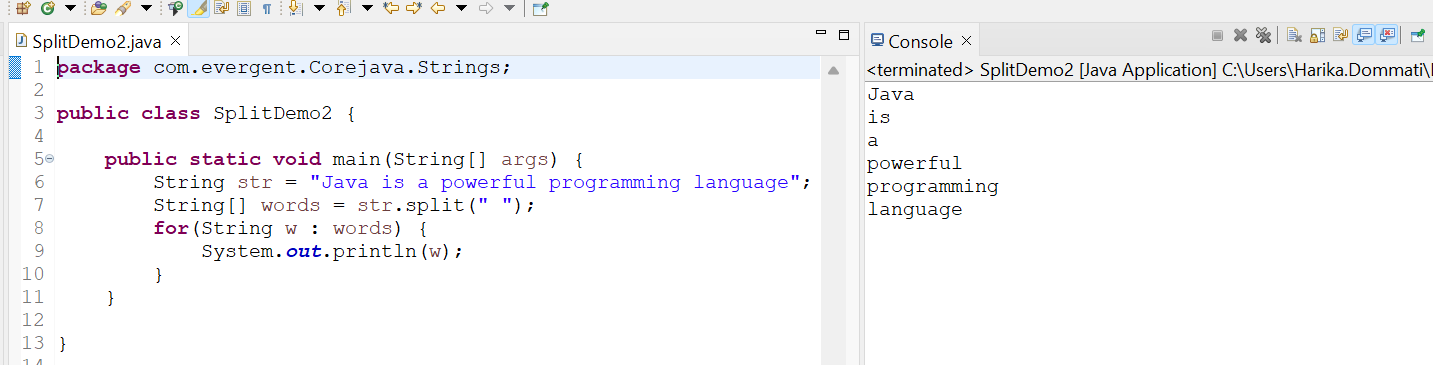
Program 9 :



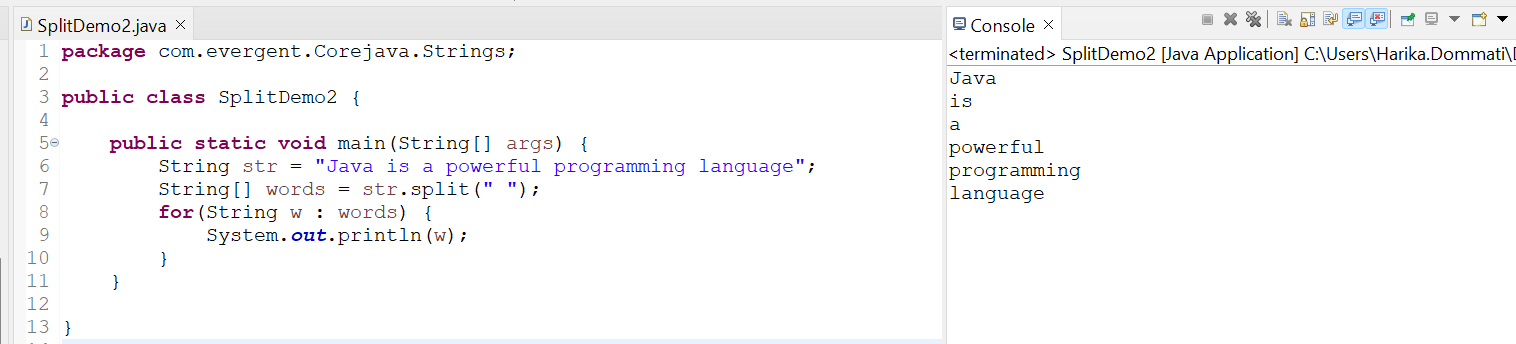
Program 10 :



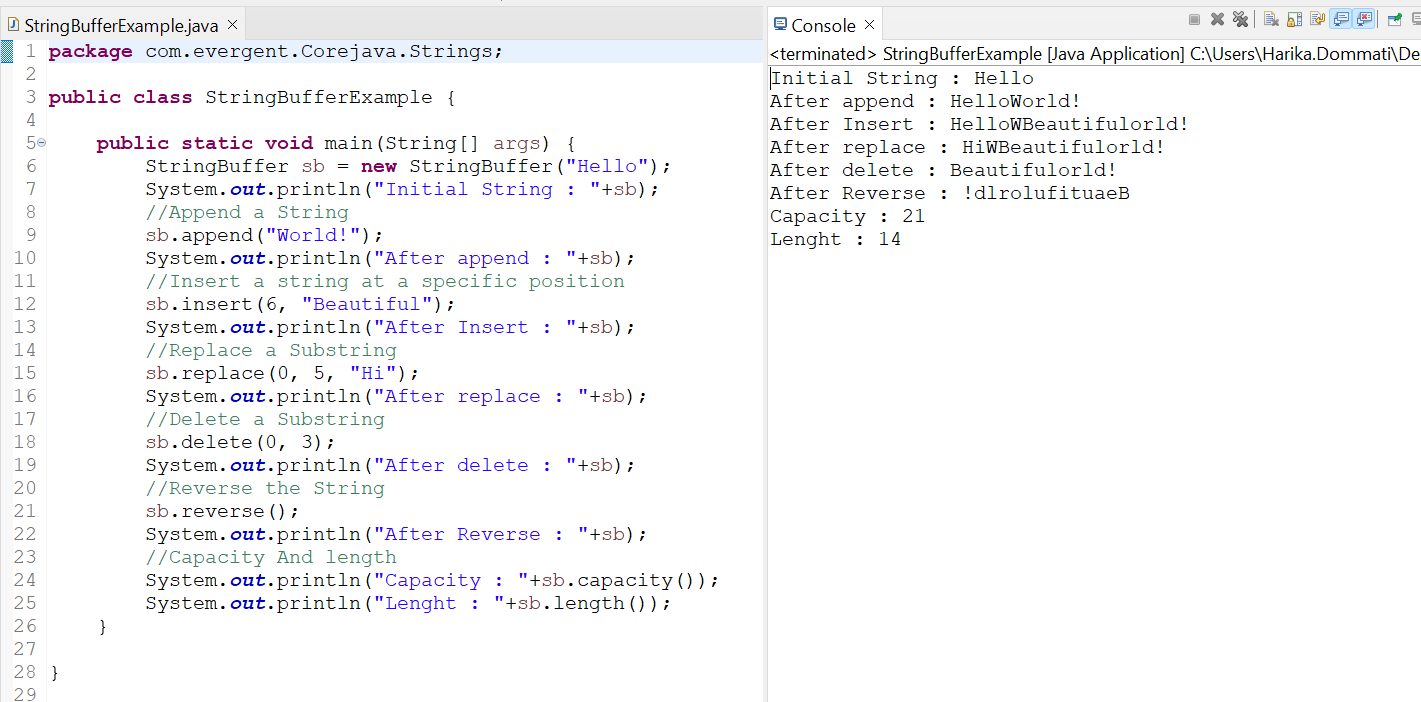
Program 11 :



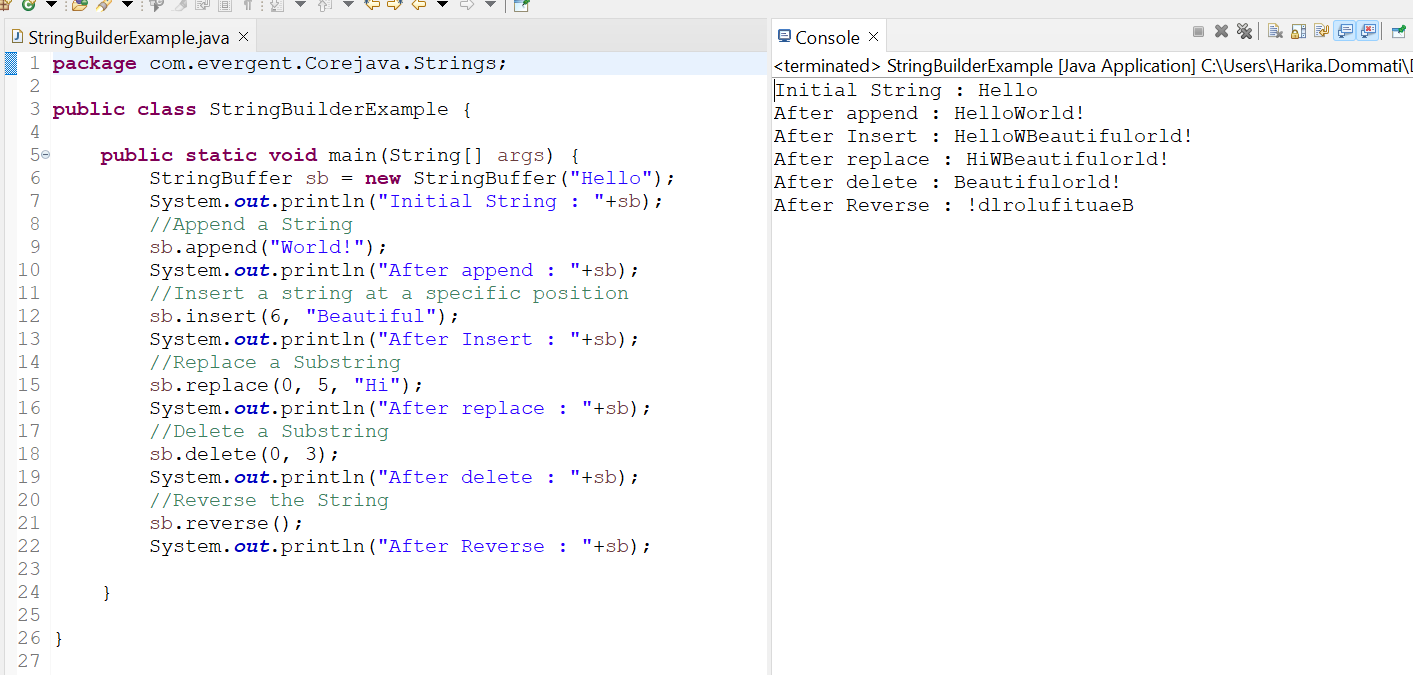
Program 12 :



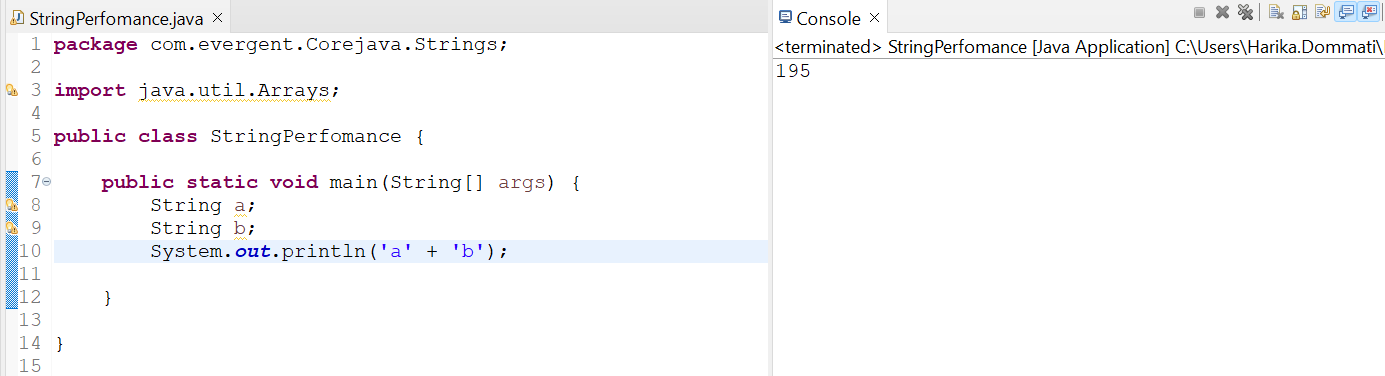
Program 13 :



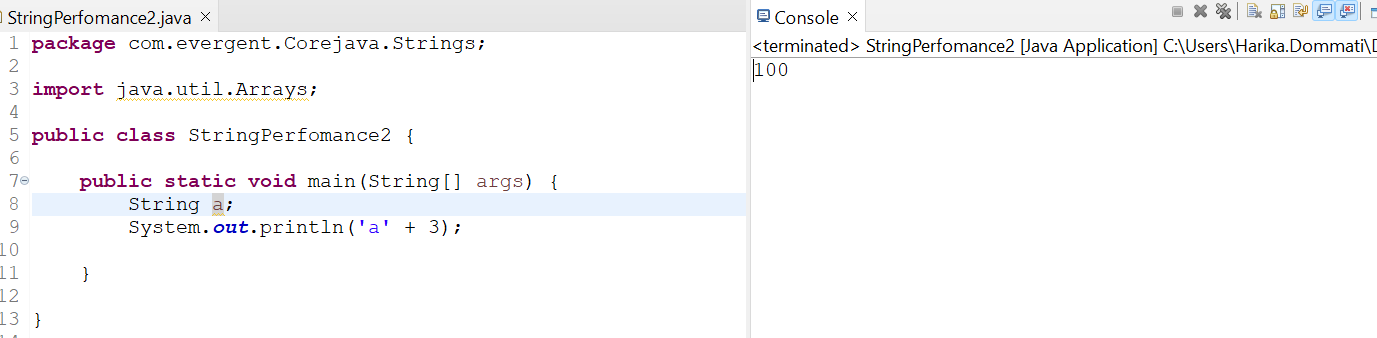
Program 14 :



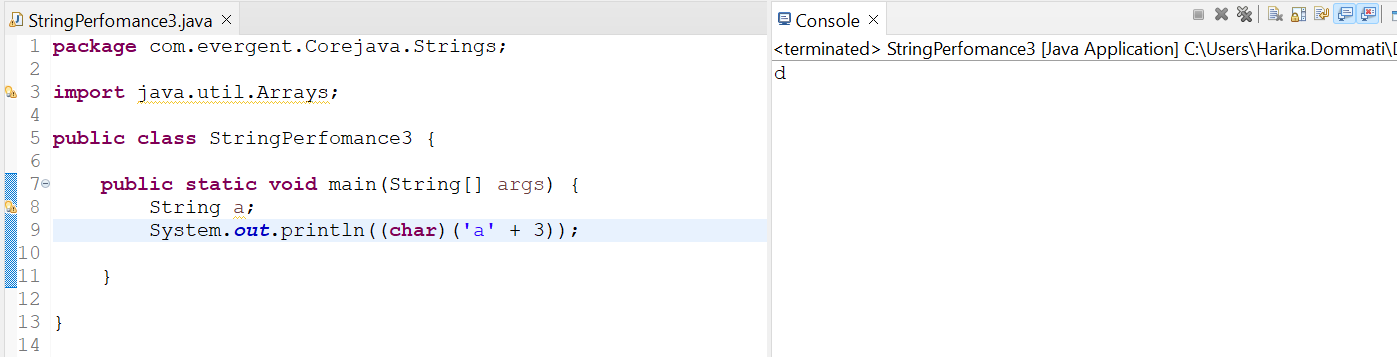
Program 15 :



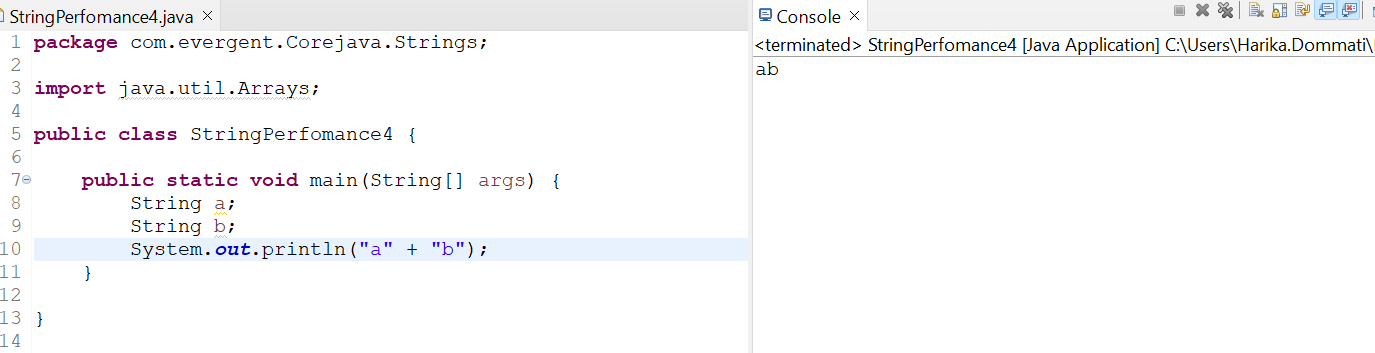
Program 16 :



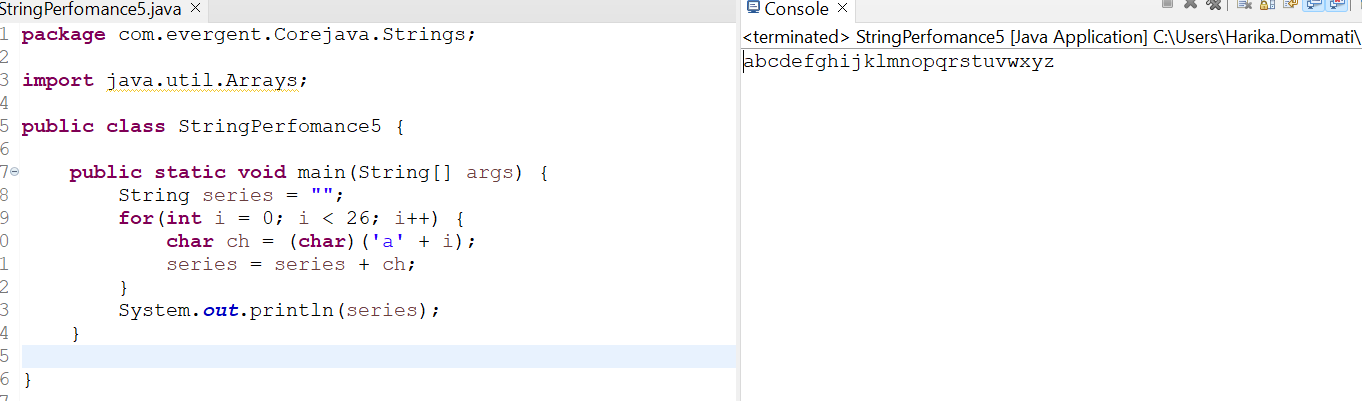
Program 17 :



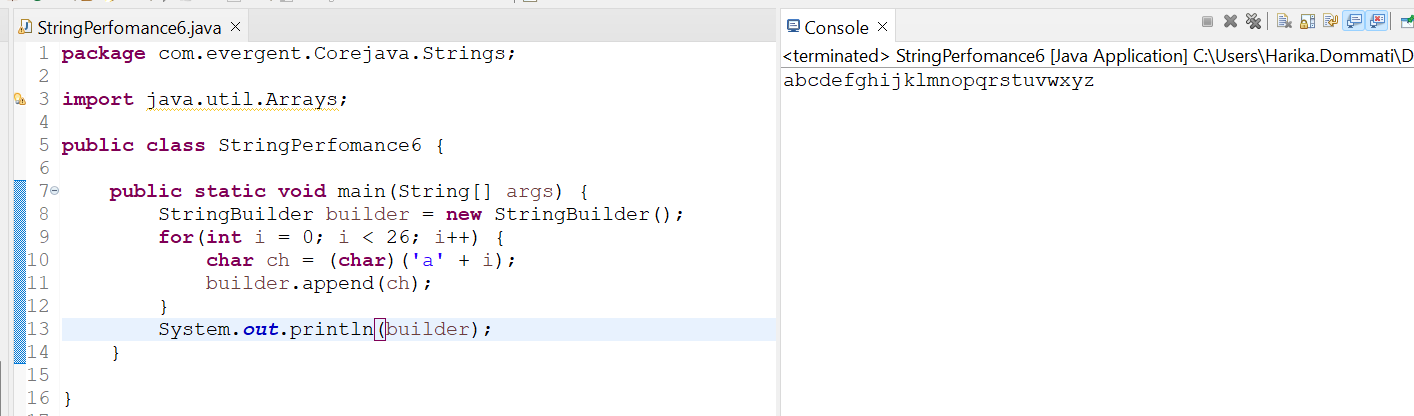
Program 18 :



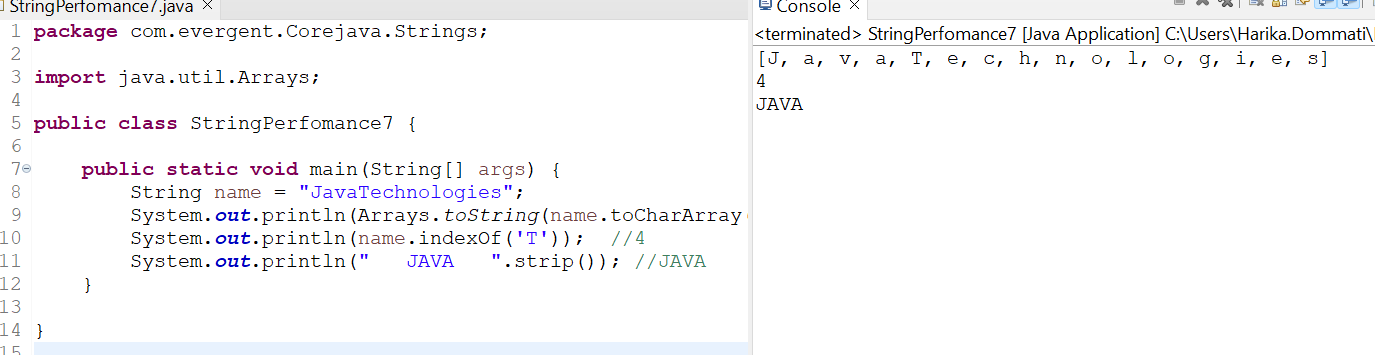
Program 19 :



Program 20 :



Program 21 :



**Day:7**

**Date:13-08-2024**

**INTERFACES:**

1. Interface is a **Keyword**

2. We can declare method signatures only but not implementation

3. By default all interfaces methods are **Abstract**

4. If any class implements interface,that class should **override all interface methods** otherwise the class is showing **compile time error.**

5. We cannot create object to interface but we can create reference to interface -*Class implements interface program 2*

6. The **variables** inside the interfaces are **public,static,final**

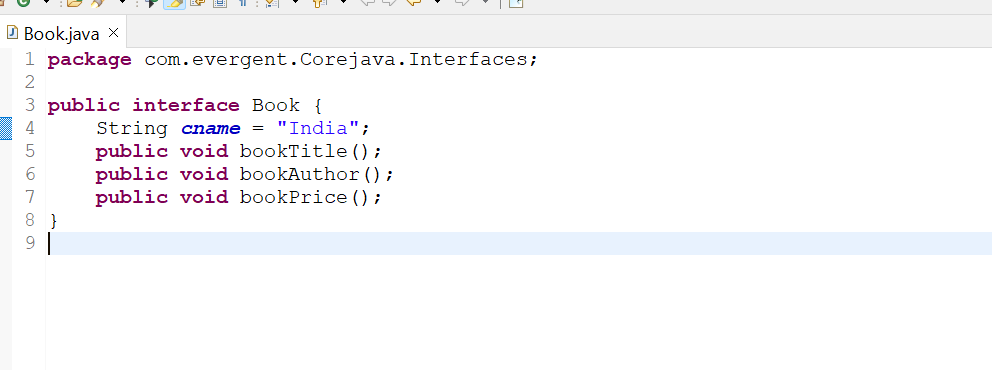
7. Java support **multiple inheritance** through **interfaces**

8. One **class** can **implements** more than one **interface**

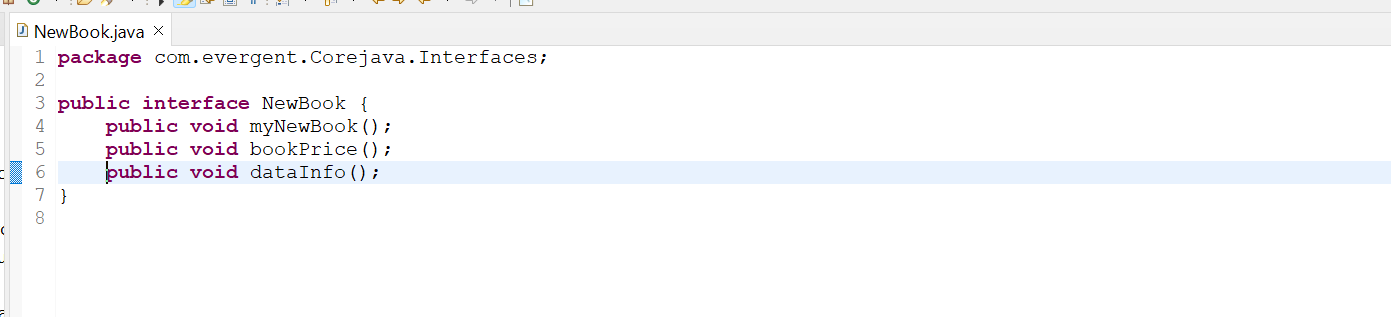
9. One **interface** can **extends** other **interfaces**

**INTERFACE PROGRAMS :**

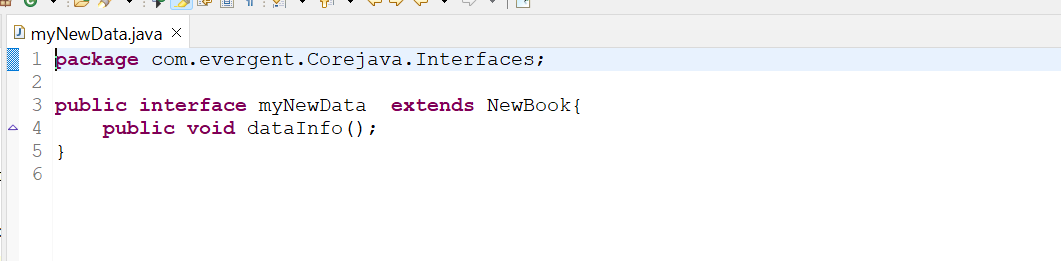
**Interface Program 1:**

****

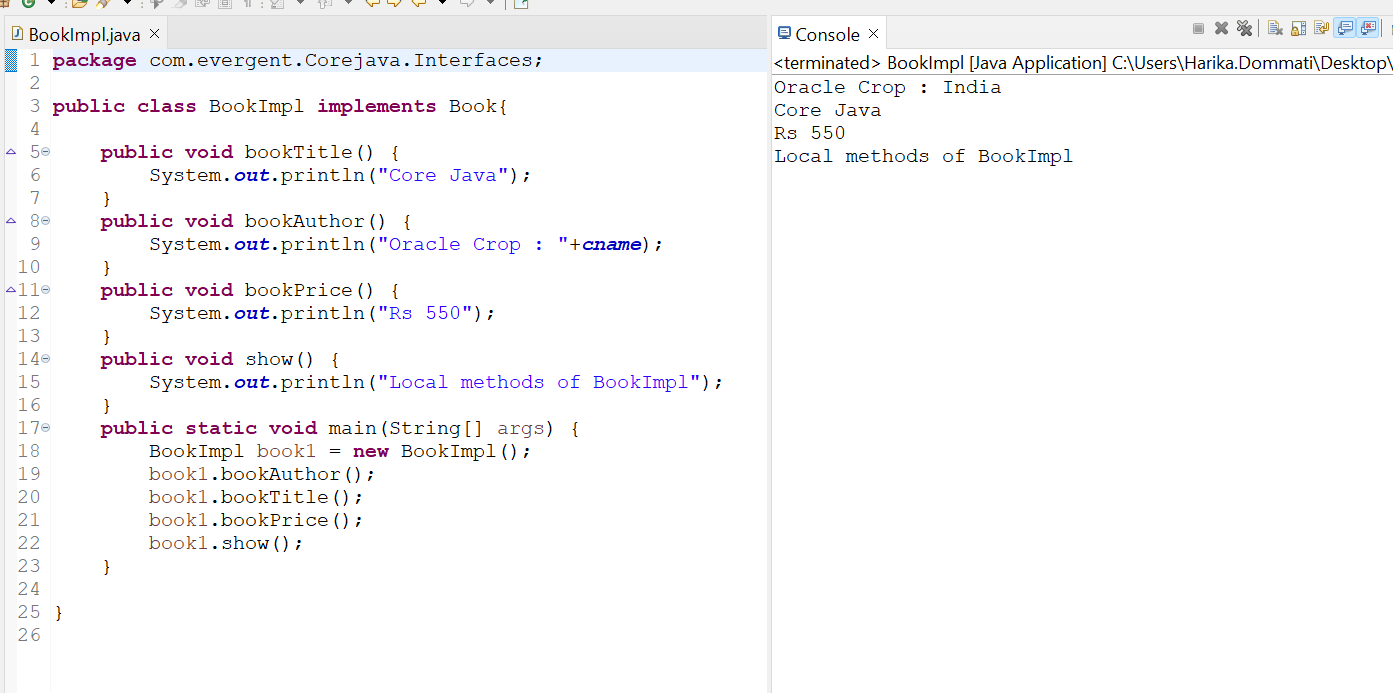
**Interface Program 2:**

****

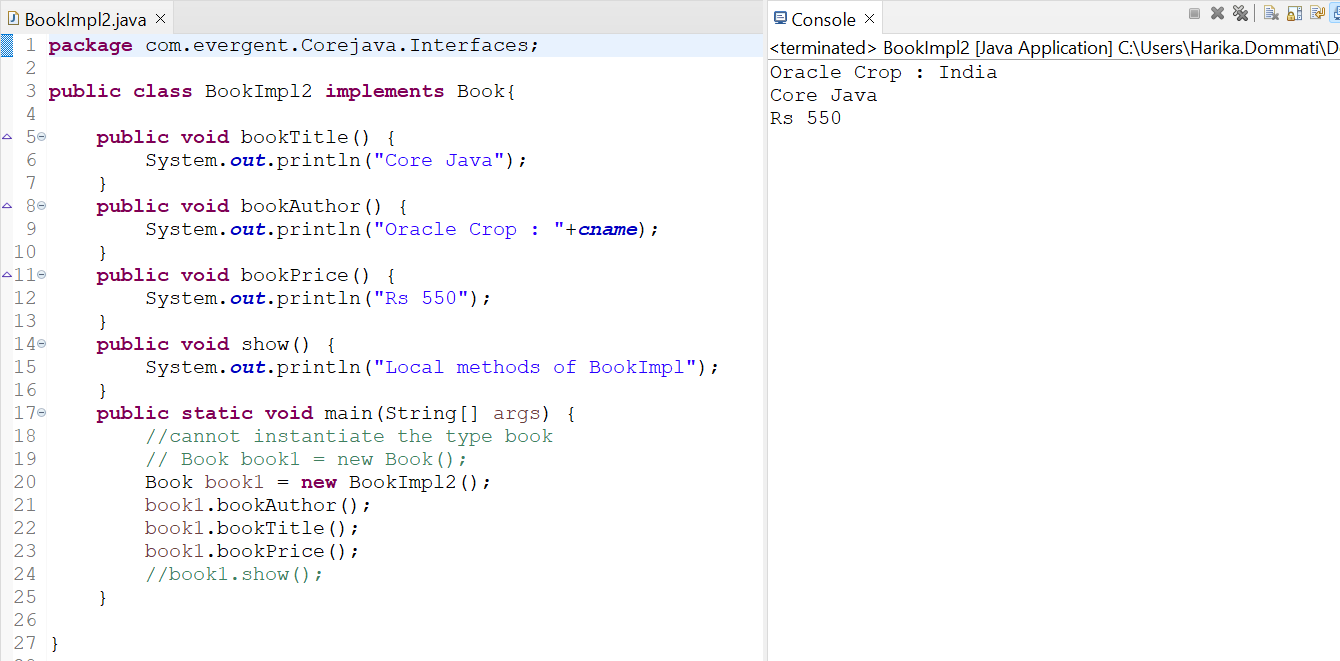
**Interface Program 3:**

****

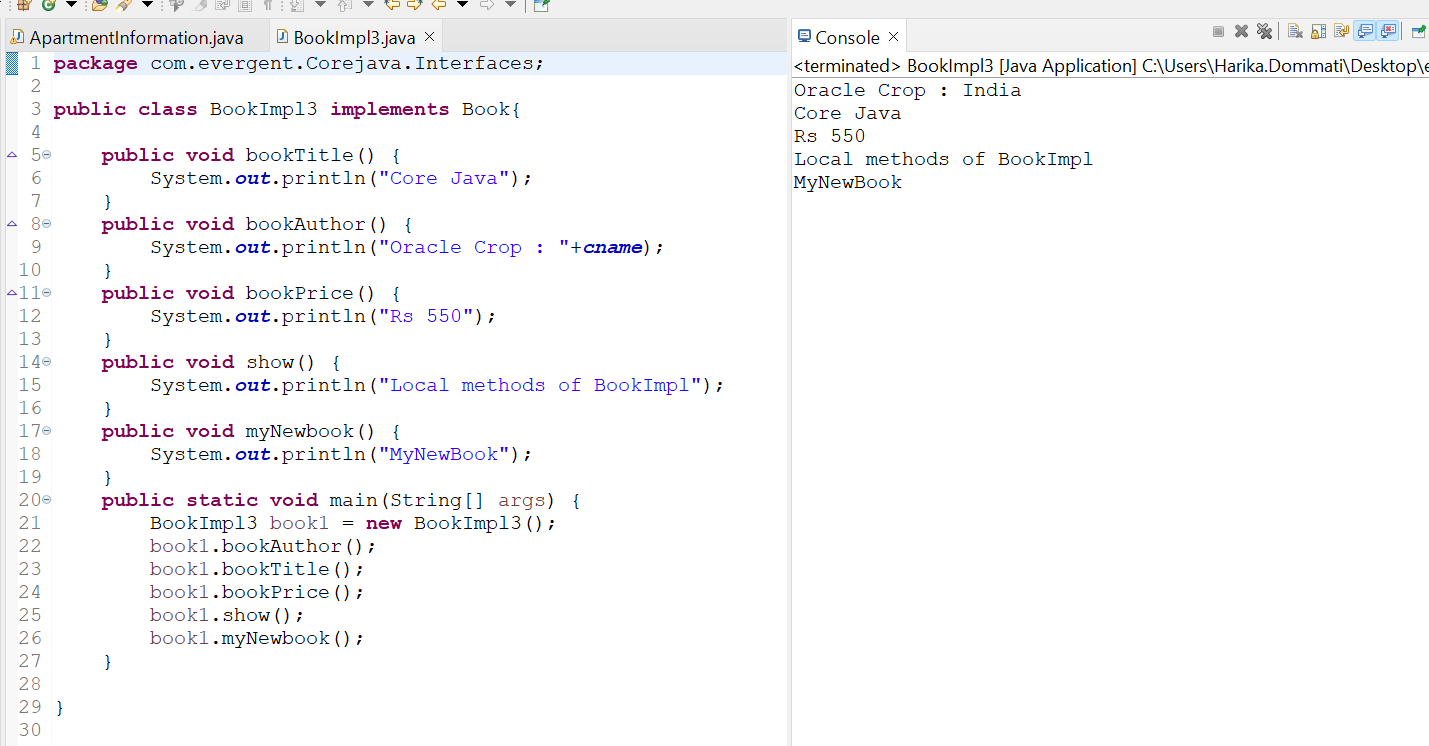
**Class Implements Interface Program 1:**

****

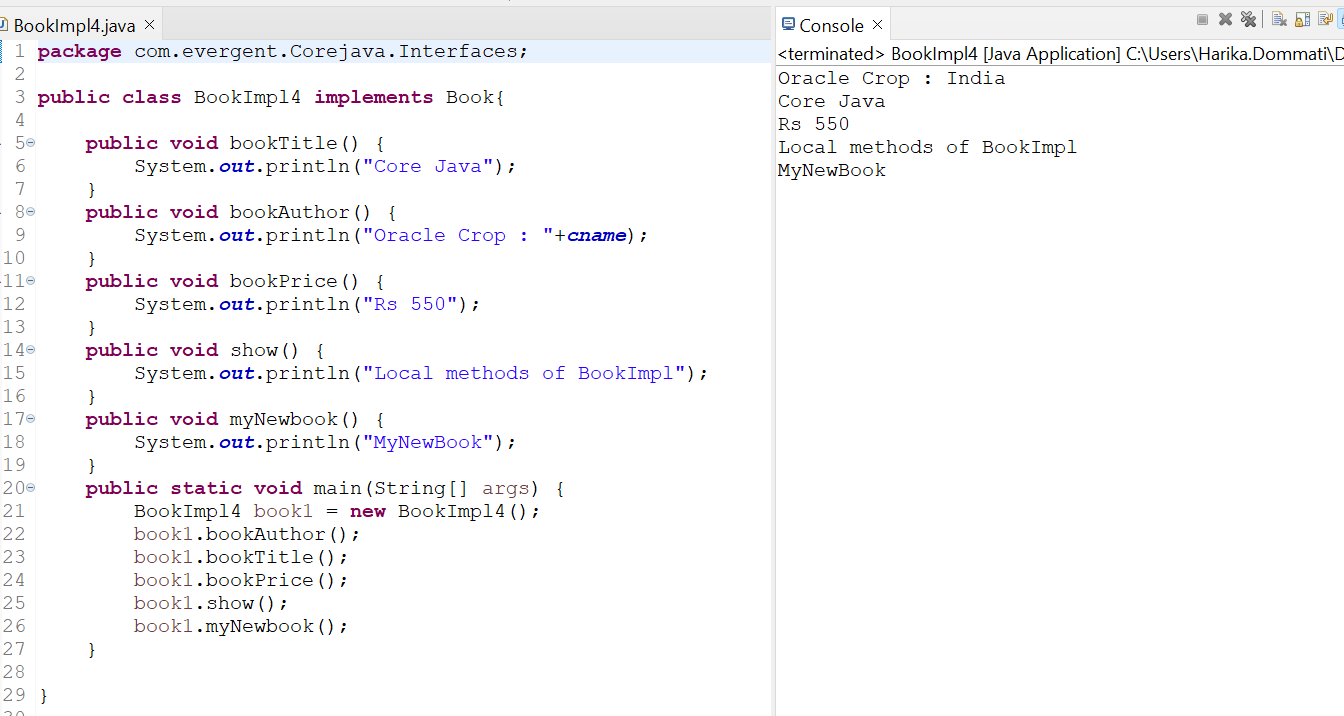
**Class implements Interface Program 2:**

****

**Class implements Interface Program 3:**

****

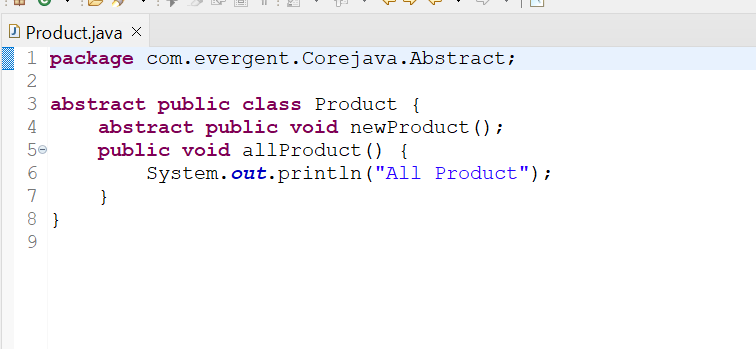
**Class implements Interface Program 3:**

****

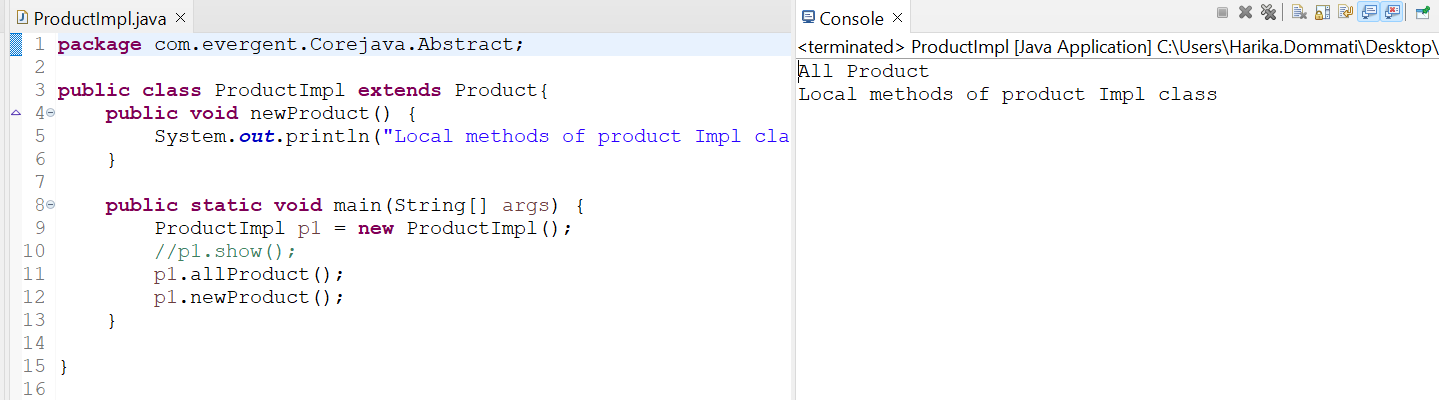
**Abstract class**

* Abstract is a keyword
* Abstract class having abstract methods and concrete(implemented) method
* If any class having one abstract method that class should be declared as a abstract keyword otherwise the class will be showing compile time error
* If any class extends abstract class that class should override all abstract methods otherwise the will be showing compile time error
* We can not create reference to abstract class but we can create reference to abstract class

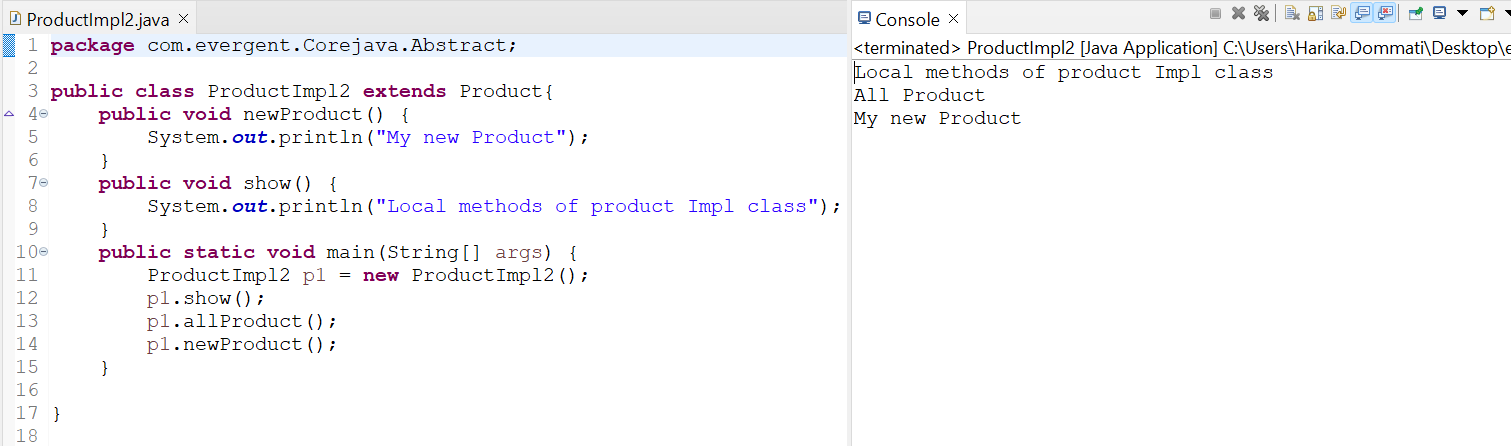
**Program 1 Abstract class :**

****

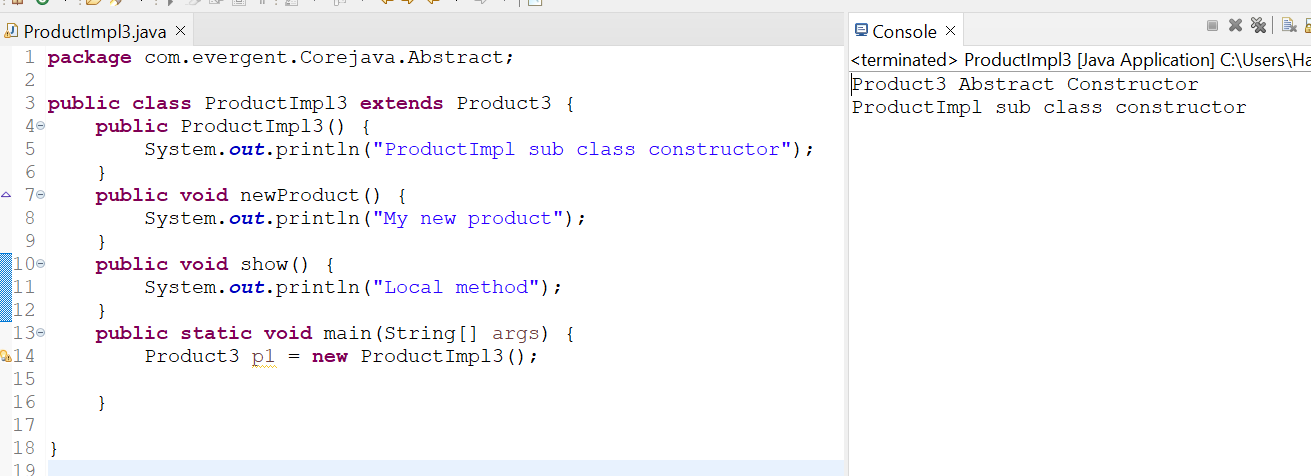
**Program 2 :**

****

**Program 3 :**

****

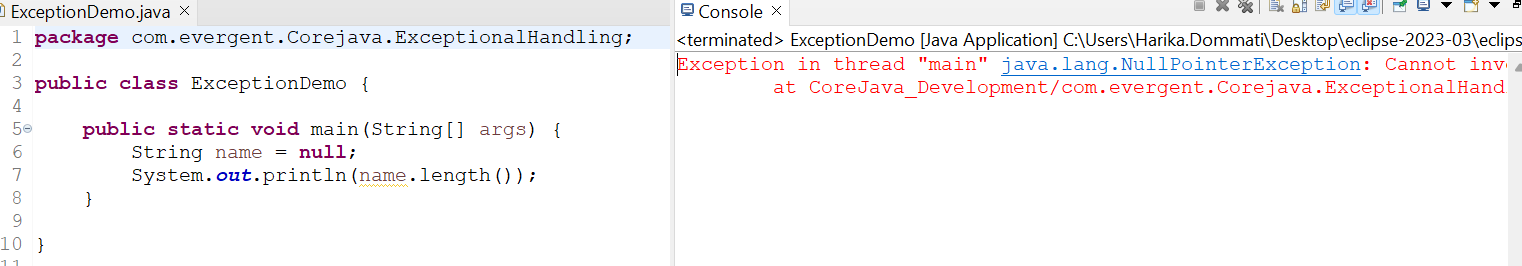
**Program 4 :**

****

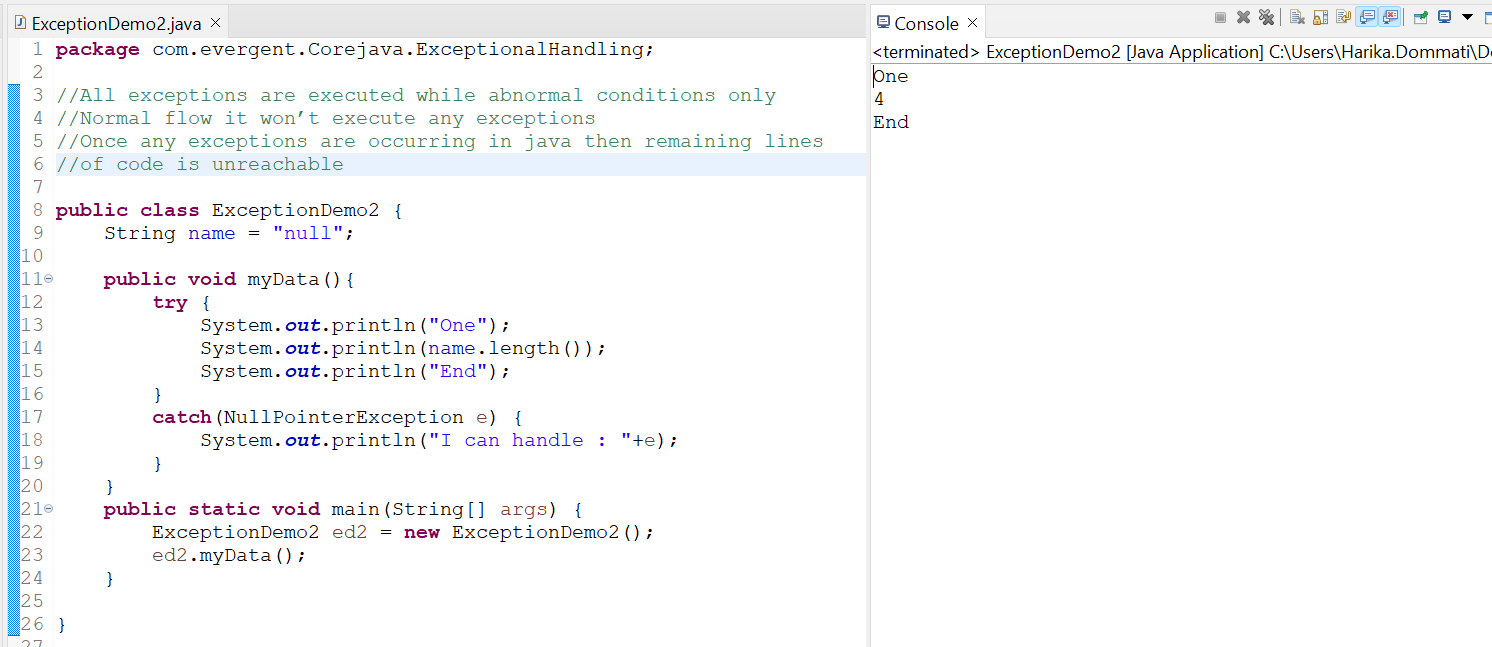
**Exceptional handling**

1. Exception handling is a mechanism
2. Exceptions are inbuilt mechanism of java
3. All exceptions are executed while abnormal conditions only
4. Normal flow it won’t execute any exceptions
5. Once any exceptions are occurring in java then remaining lines of code is unreachable
6. java.lang .Throwable is super class for exception and errors
7. There are two types of exceptions in java
   1. Checked Exception
   2. Unchecked Exception
8. All checked exceptions are compile time exceptions
9. All Unchecked exceptions are runtime exceptions
10. There are 5 keywords in Exceptions Handling
    1. Try
    2. catch()
    3. finally()
    4. Throws
    5. Throw
11. Try is for business logic
12. Catch is for handling exceptions
13. Finally is a block, if exceptions is occur or not finally block will be executed
14. throws an exception will be executed method by method
15. throw is for both compile time and run time exceptions and will call predefined exceptions or user defined exceptions
16. try followed by either catch block or finally block
17. We should follow exceptions hierarchical
18. We can create our own(User Defined) exceptions
19. Our own exceptions extends exception or runtime exception
20. All exceptions classes are in to java.lang package
21. There is two exceptions in class,developer should be handle one after one
22. Developer can’t handle errors

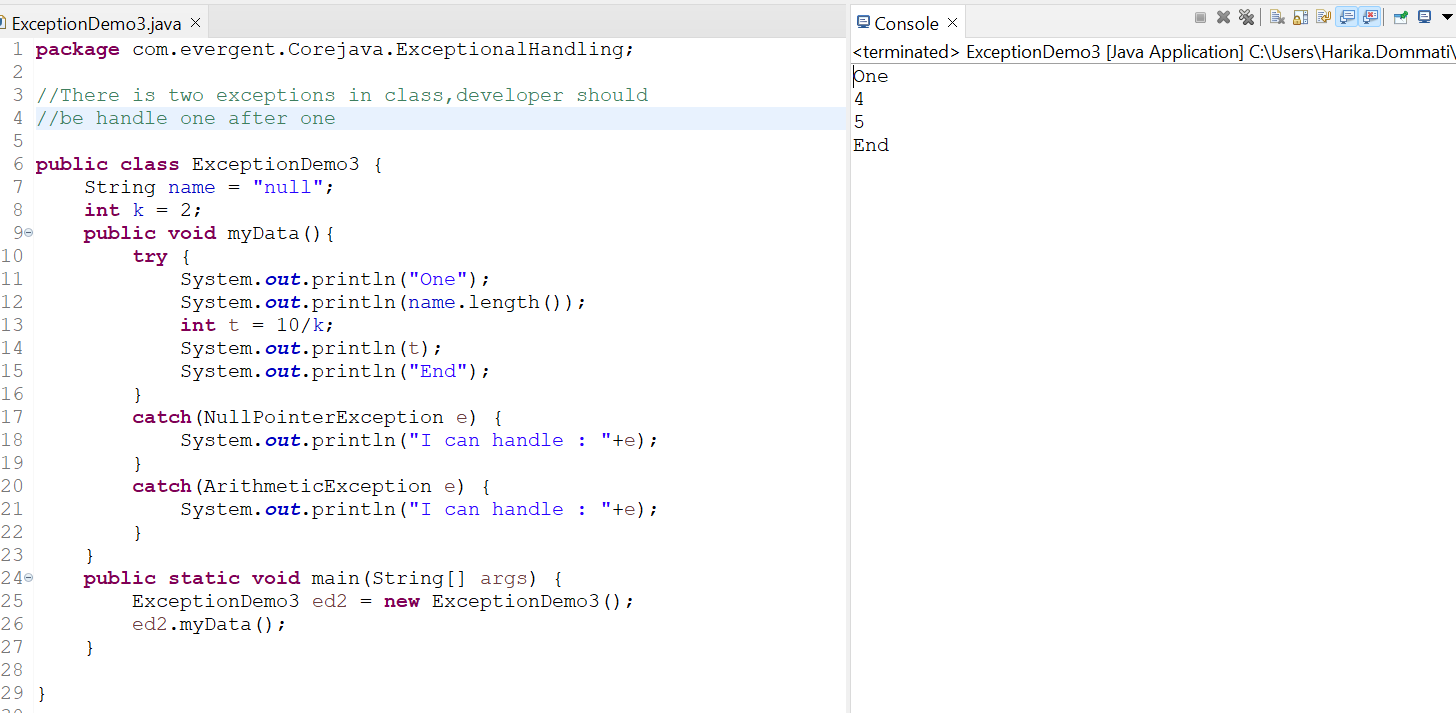
**Program 1 :**



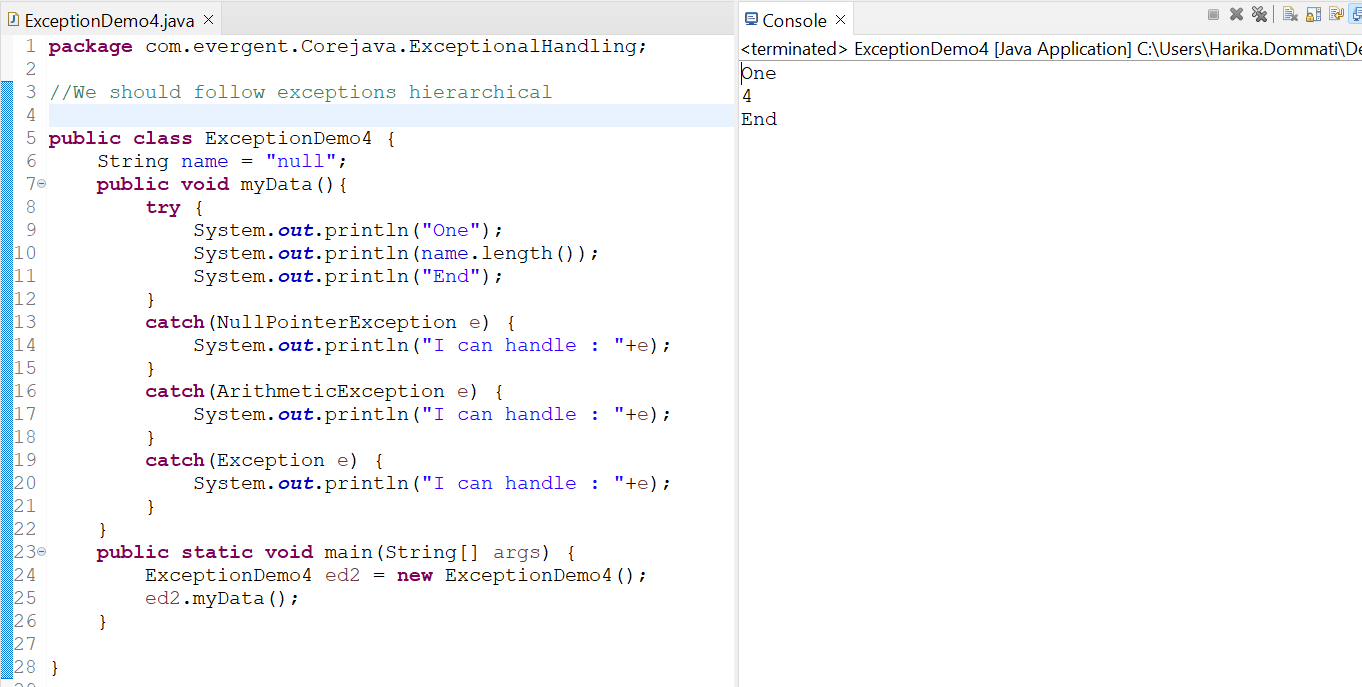
**Program 2 :**

****

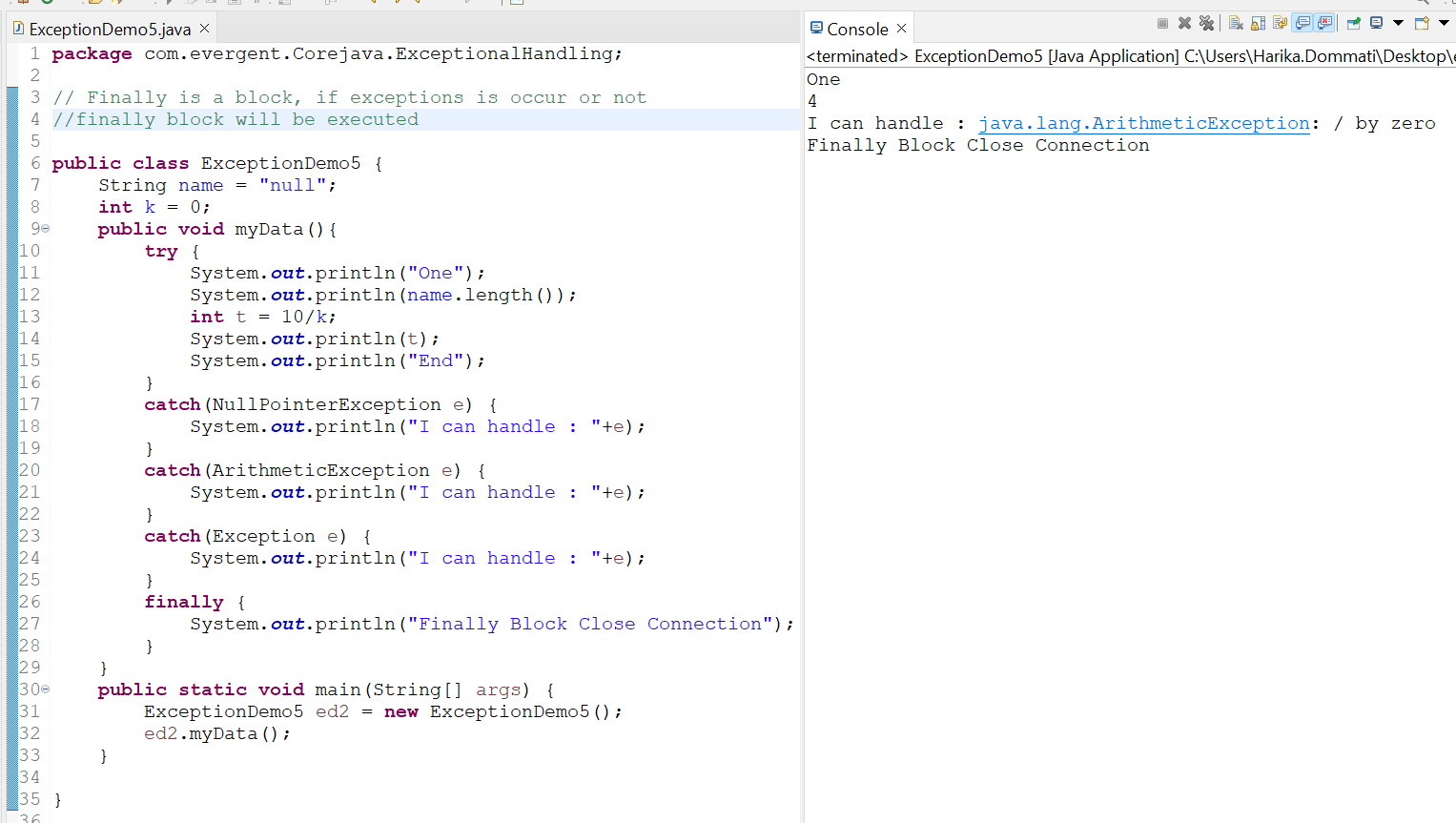
**Program 3 :**

****

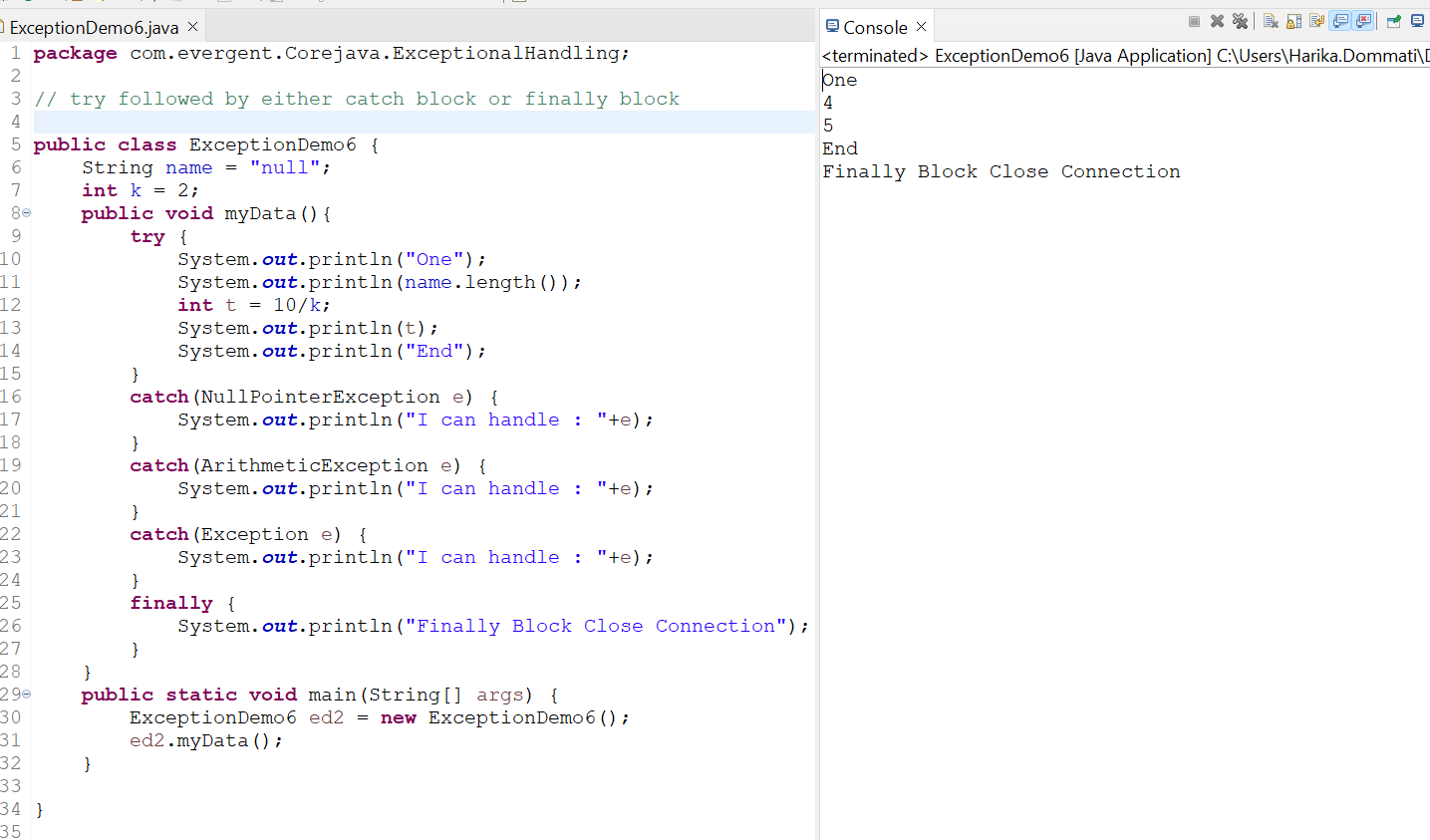
**Program 4 :**

****

**Program 5 :**

****

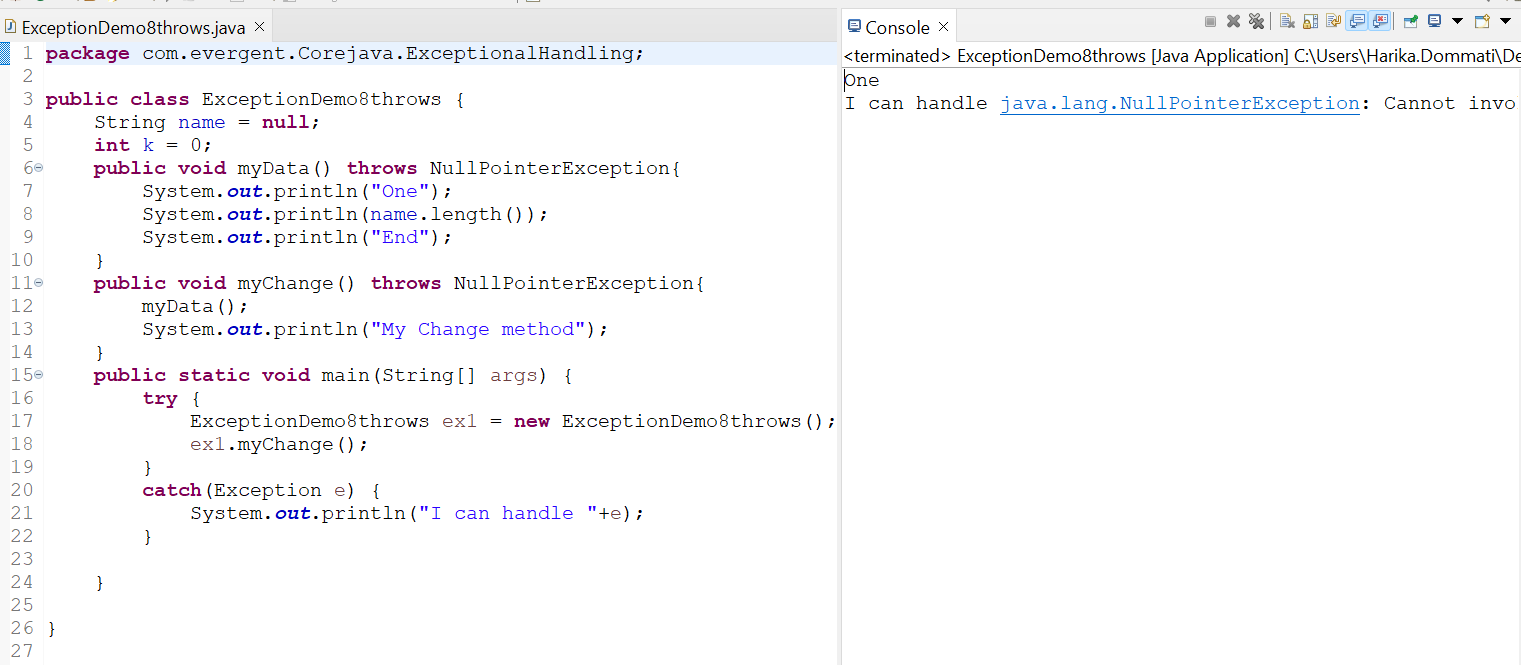
**Program 6 :**

****

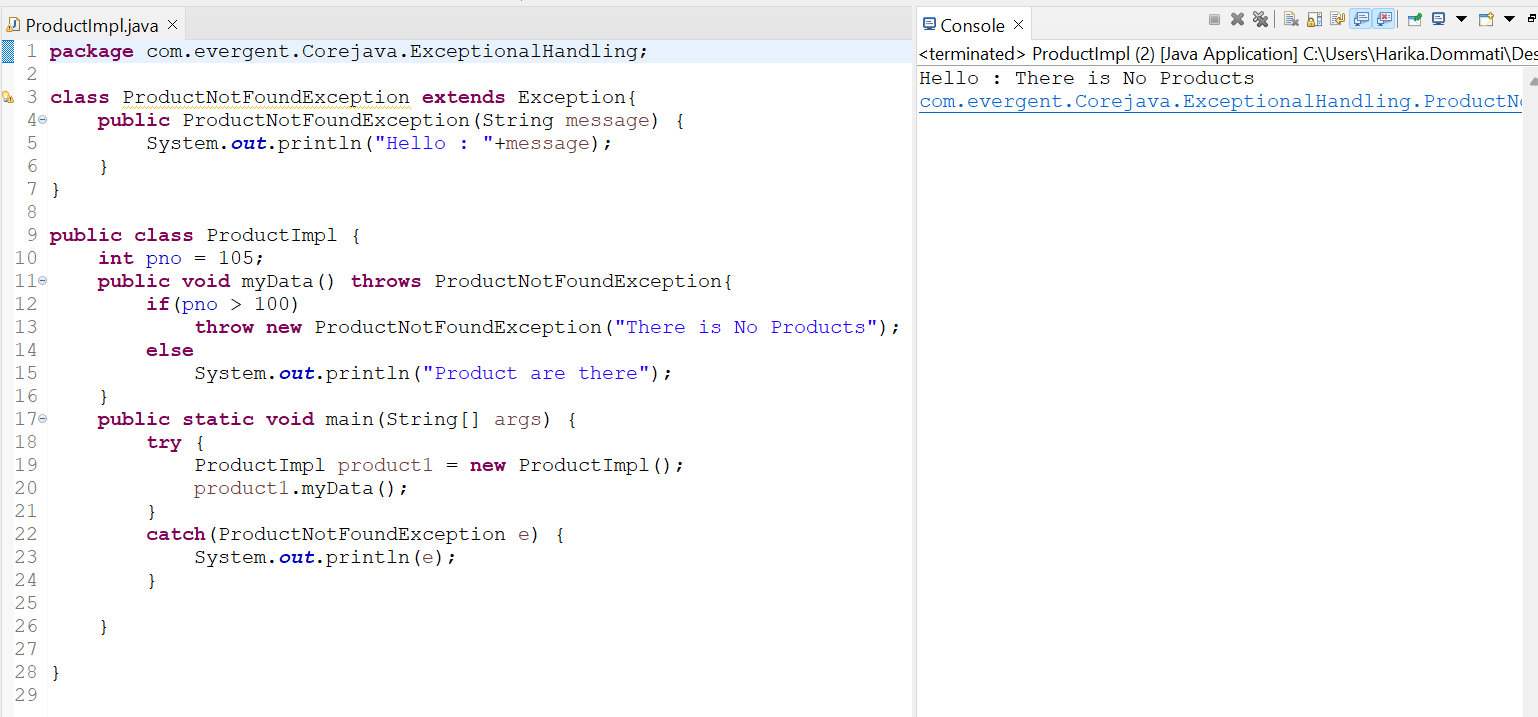
**Program 7 :**

****

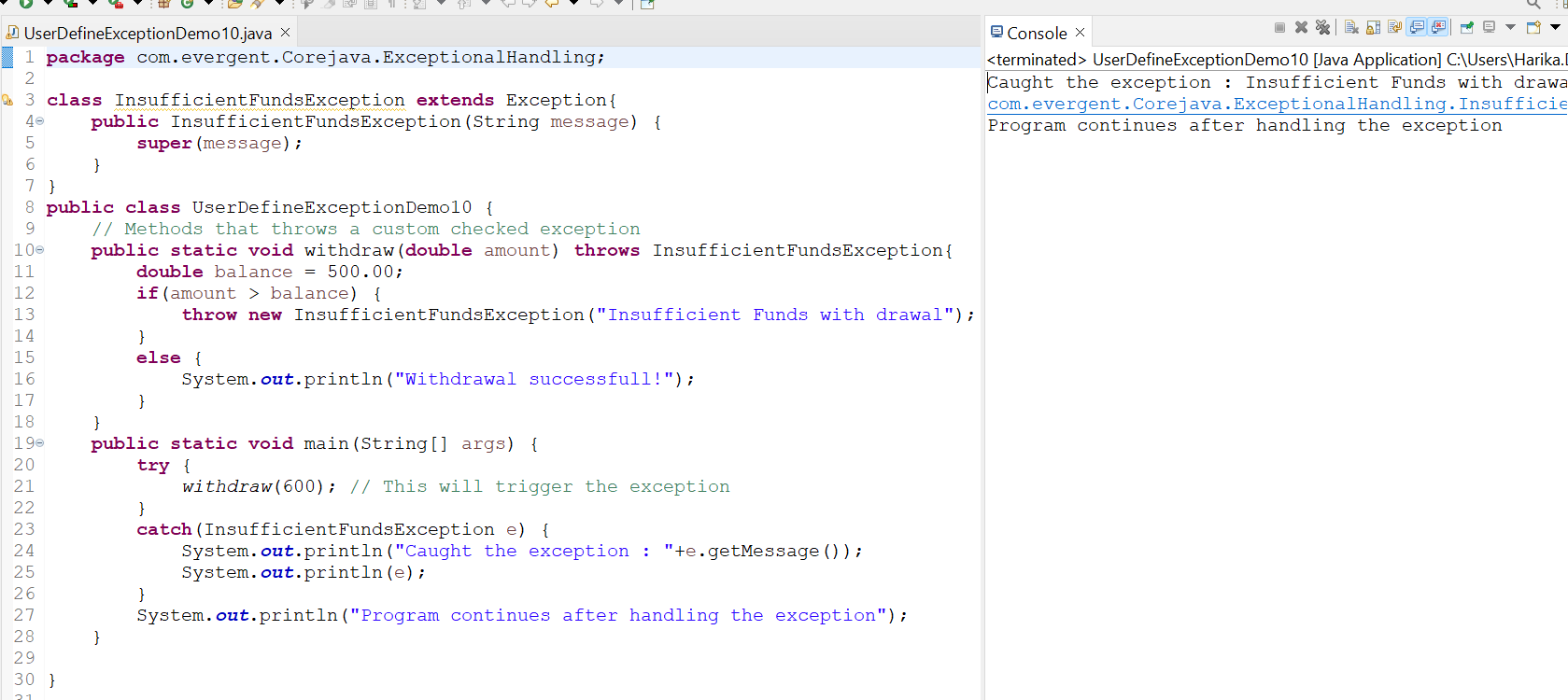
**Program 8 :**

****

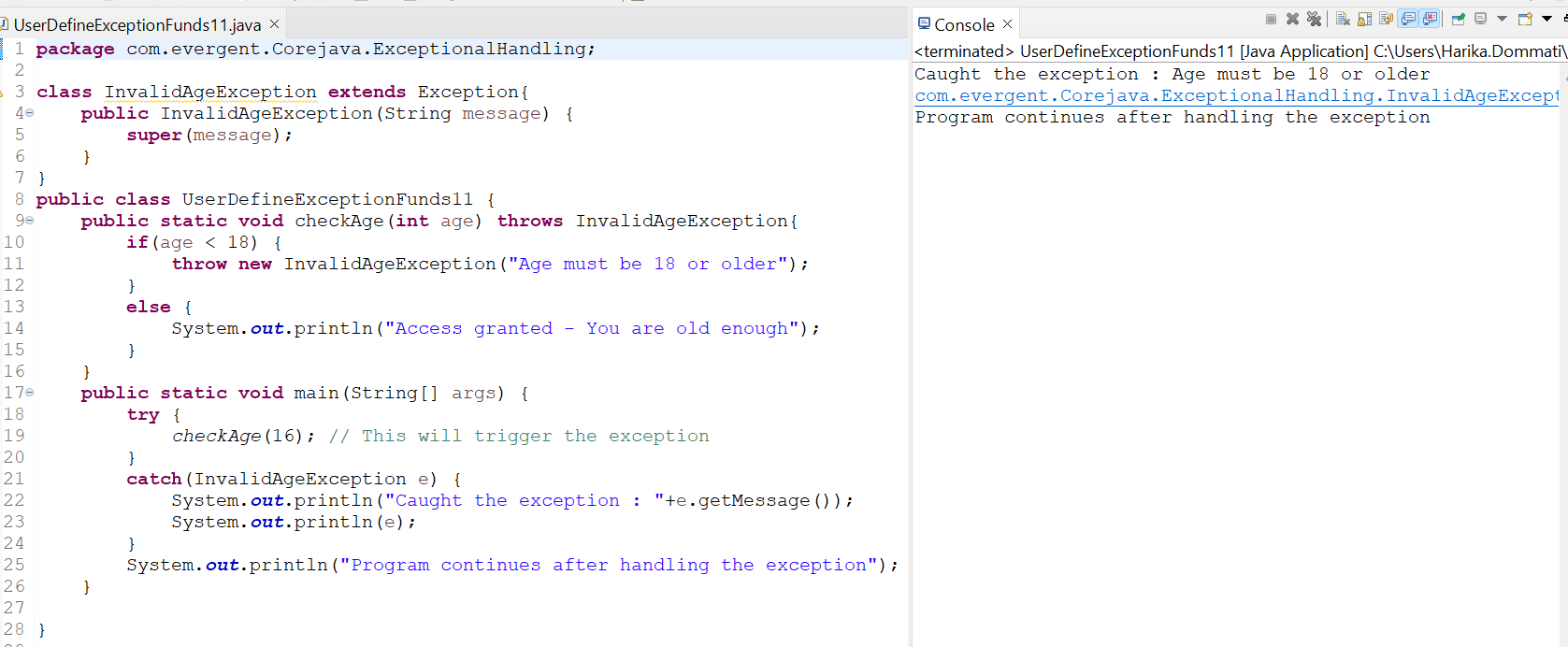
**Program 9 :**

****

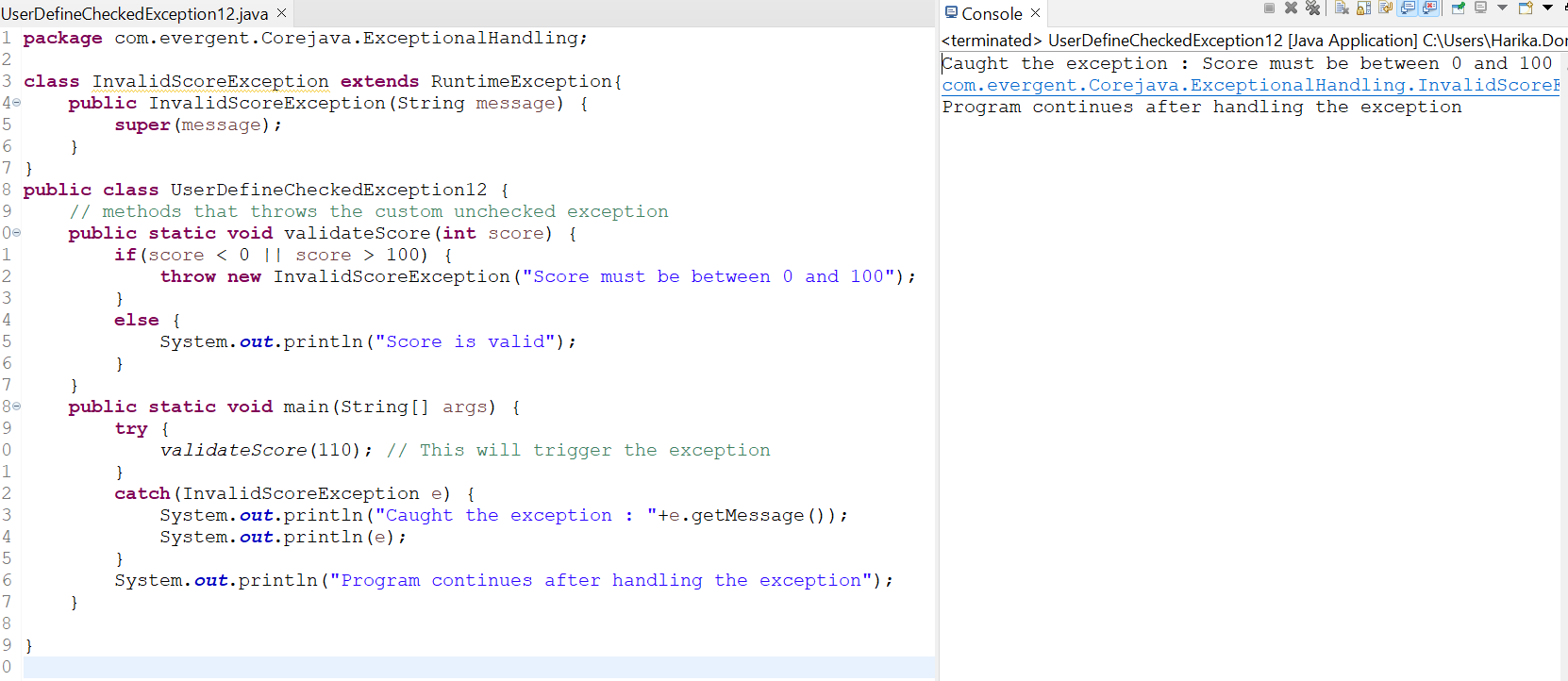
**Program 10:**

****

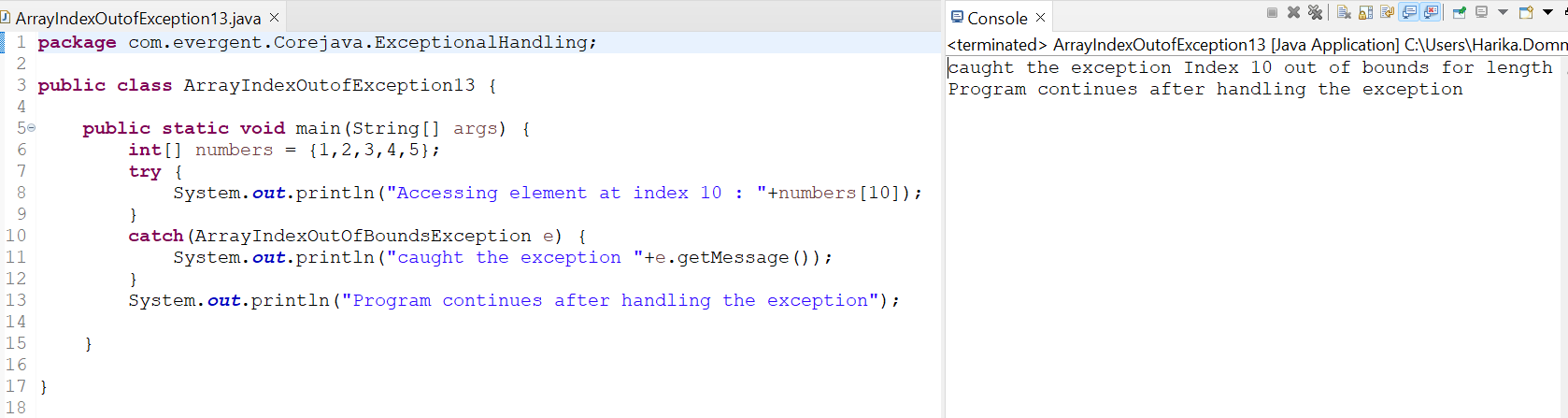
**Program 11 :**

****

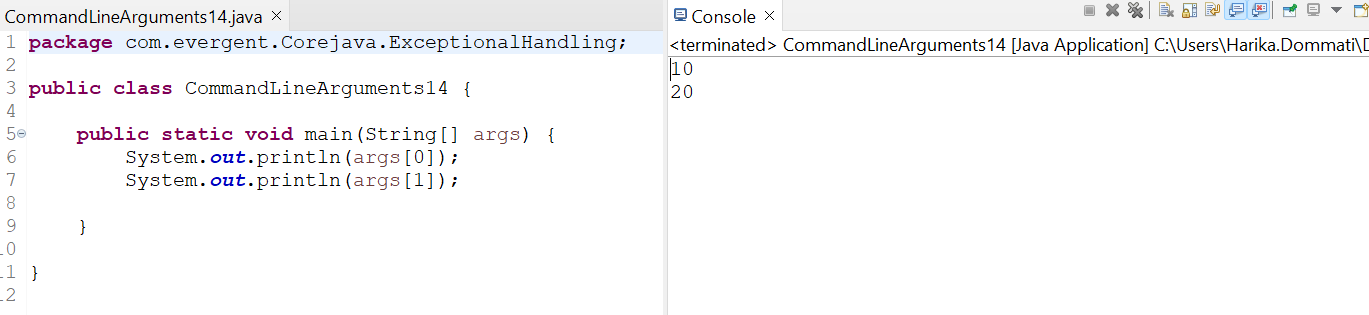
**Program 12 :**

****

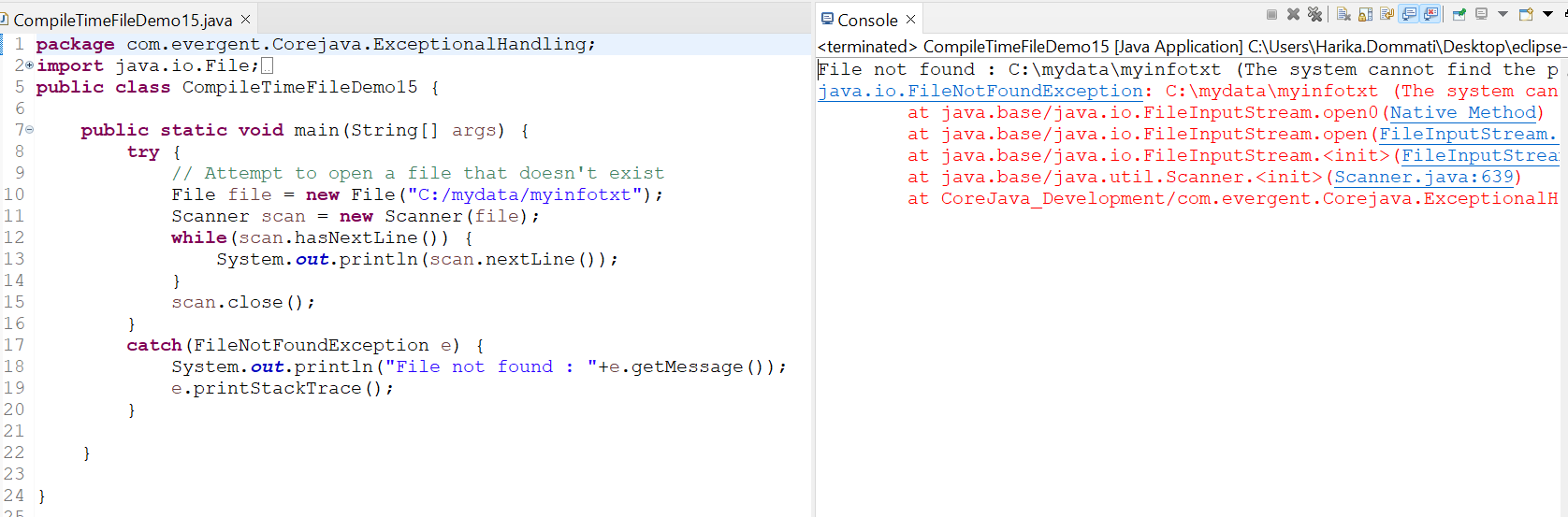
**Program 13 :**

****

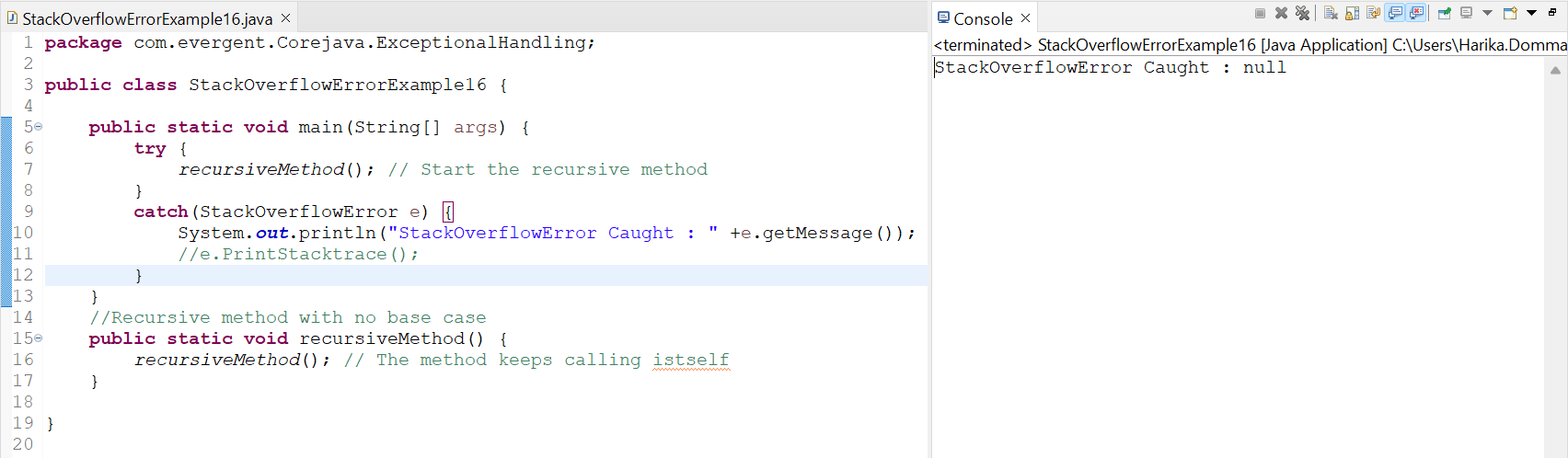
**Program 14 :**

****

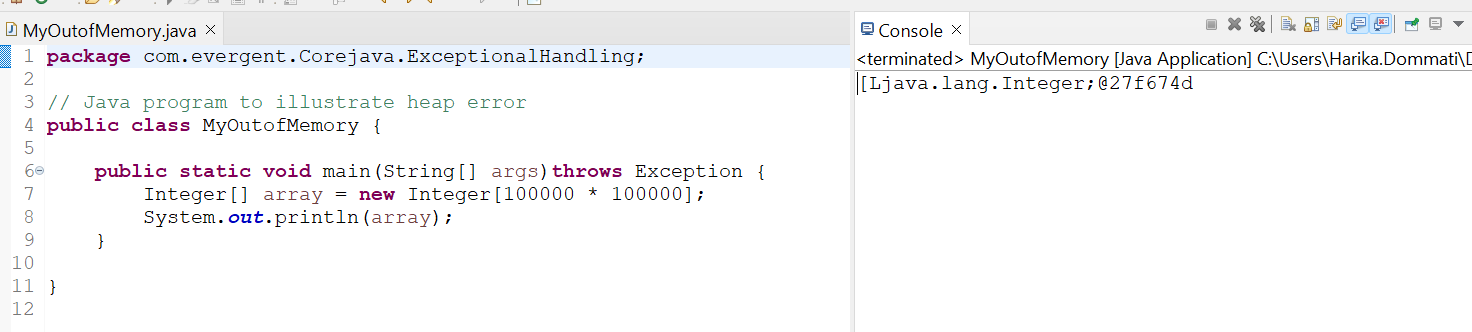
**Program 15 :**

****

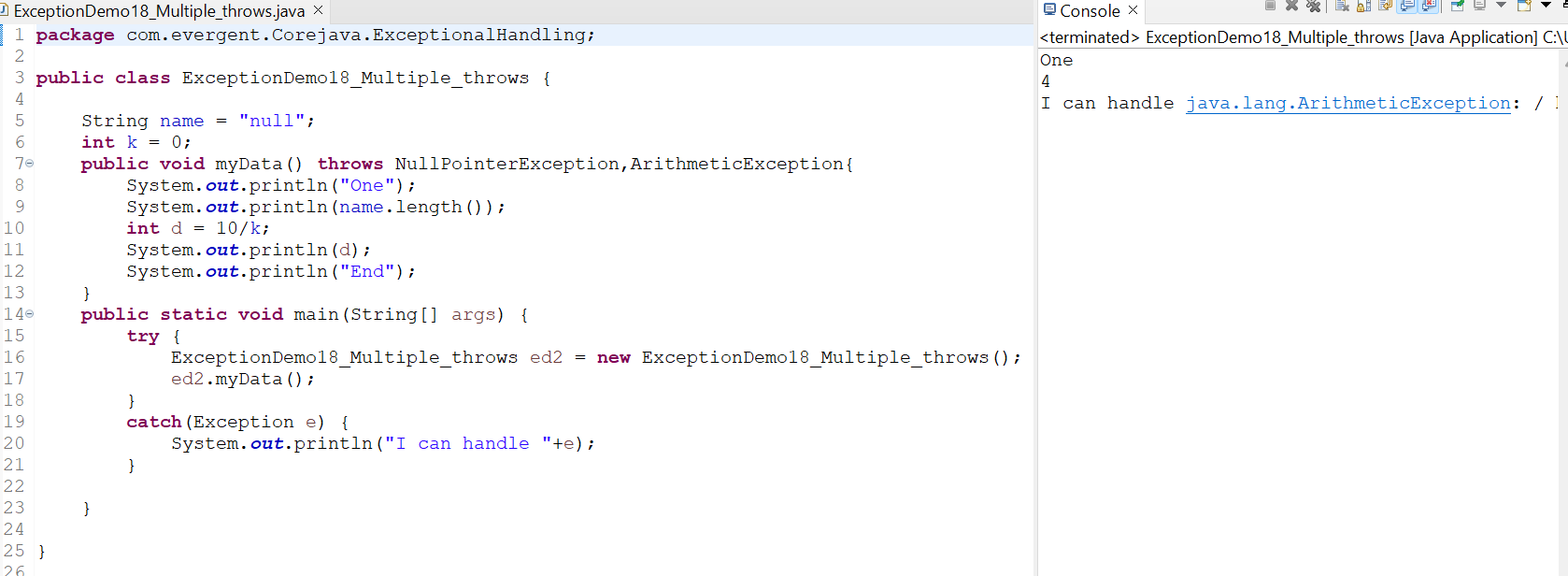
**Program 16 :**

****

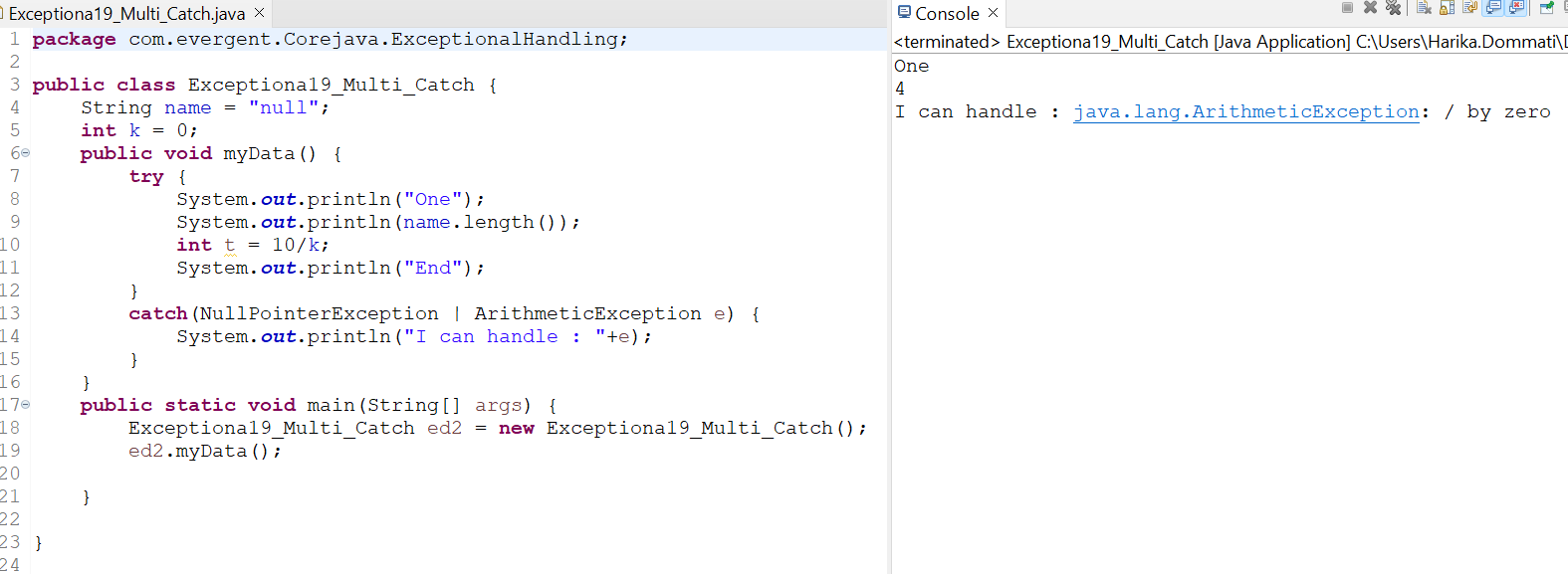
**Program 17 :**

****

**Program 18 :**

****

**Program 19 :**

****