**Core Java Index** (Version 1)

**Day-1 (05-08-2024)**

1. Language and Applications
2. Java Features
3. Why Java is Platform Independent
4. Oops
5. Exception Handling
6. Multi-Threading
7. Web applications
8. Open source
9. Security
10. Supports Networking
11. Memory Management
12. JDK, JRE, JVM
13. Conditional Statements ,Operators
14. Basic Java Programming
15. Largest of Two numbers
16. Largest of Three numbers
17. Swapping of two numbers with and without temp variable
18. Packages
19. Practice program: Largest of 5 numbers

**Day -2 (06-08-2024)**

1. Nested loops- Pattern Programs
2. Arrays - 1D example programs
3. 2D Arrays - example programs
4. Logical Programmings
5. Switch cases
6. Java.lang package & object class methods
7. Scanner
8. Enumeration
9. Event management Applications

**Day -3 (07-08-2024)**

1. Oops
2. Encapsulation
3. Calculations
4. PersonClass
5. MethodFlow
6. System.Out.Println()
7. Inheritance
8. Polymorphism
9. MethodOverloading
10. MethodOverriding
11. Abstract
12. IS-A Relation (inheritance )
13. HAS-A Relation(Object creation)

**Day -4 (08-08-2024)**

**Constructors**

-> Class name and Constructor name should be Same

-> They are 2 types of Classes

1. Default constructors
2. Parameterized constructors

-> We can Access constructors while creation of objects

-> Constructors are mainly for initializing.

-> Constructors doesn’t have any returntype not even void also.if we declare as a void the compiler will consider it as a method not as a constructors.

-> Every class needs atleast one Constructors.

->Always constructors are overloaded.

->this and super keyword.

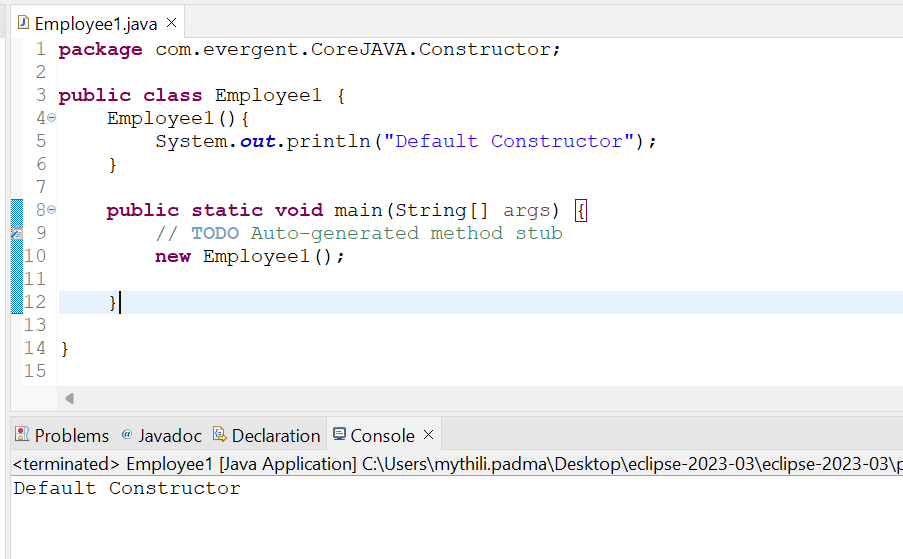
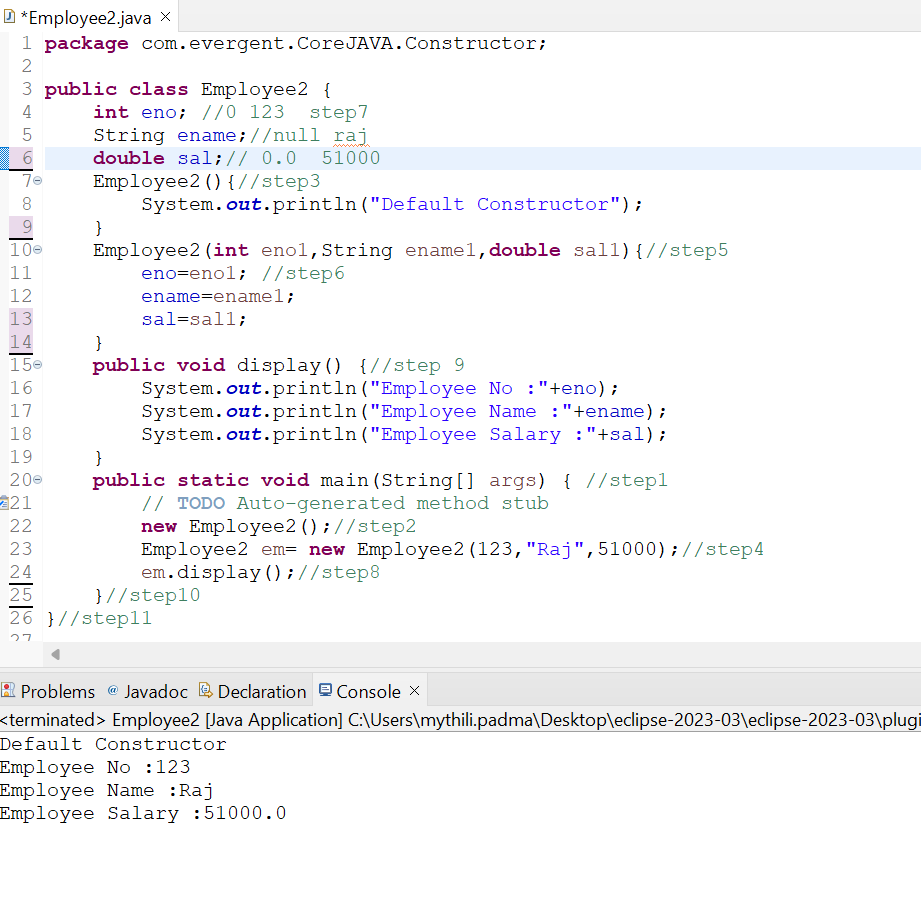
this---- keyword is always pointing to instance variable.

this can be use to call one constructor to another constructor (within a class)

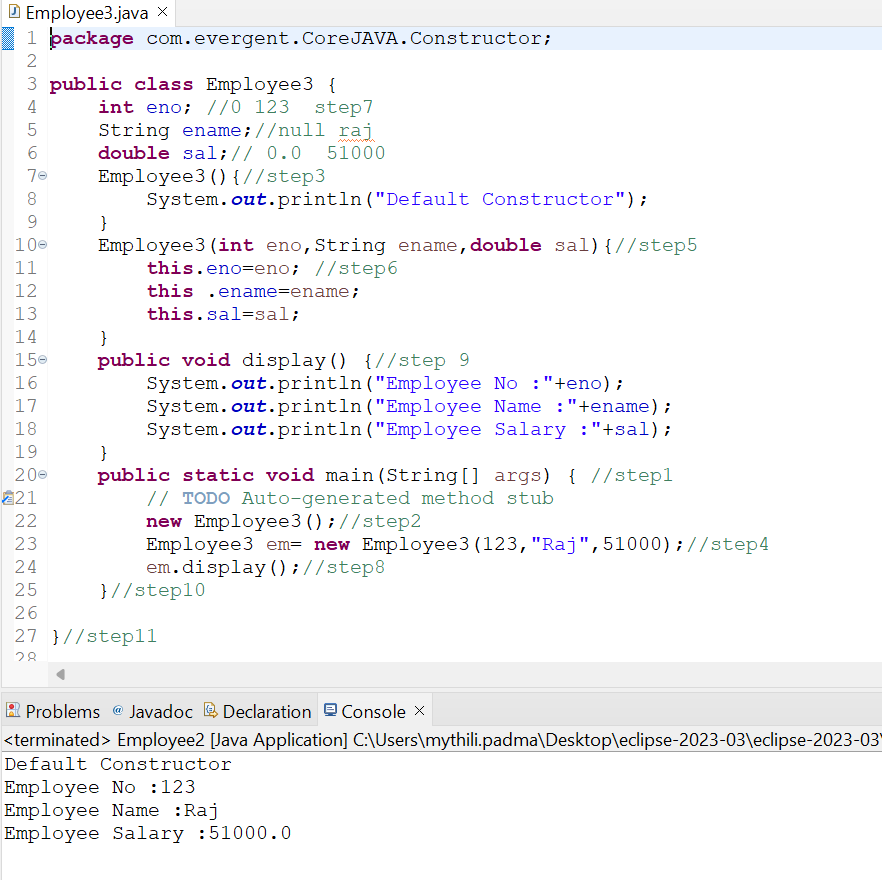
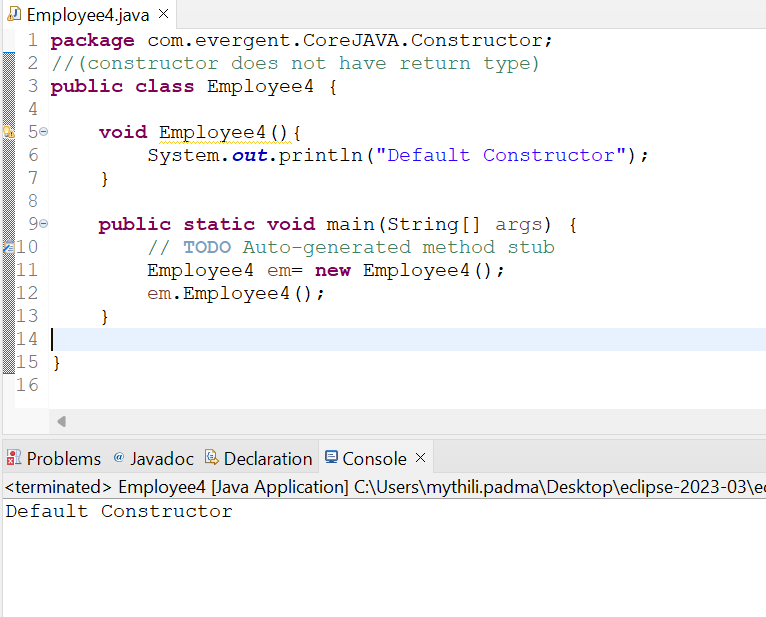
Super----- Keyword always it will call super class constructor

-> Copy Constructor ----- Object reference is copied & calls the constructor

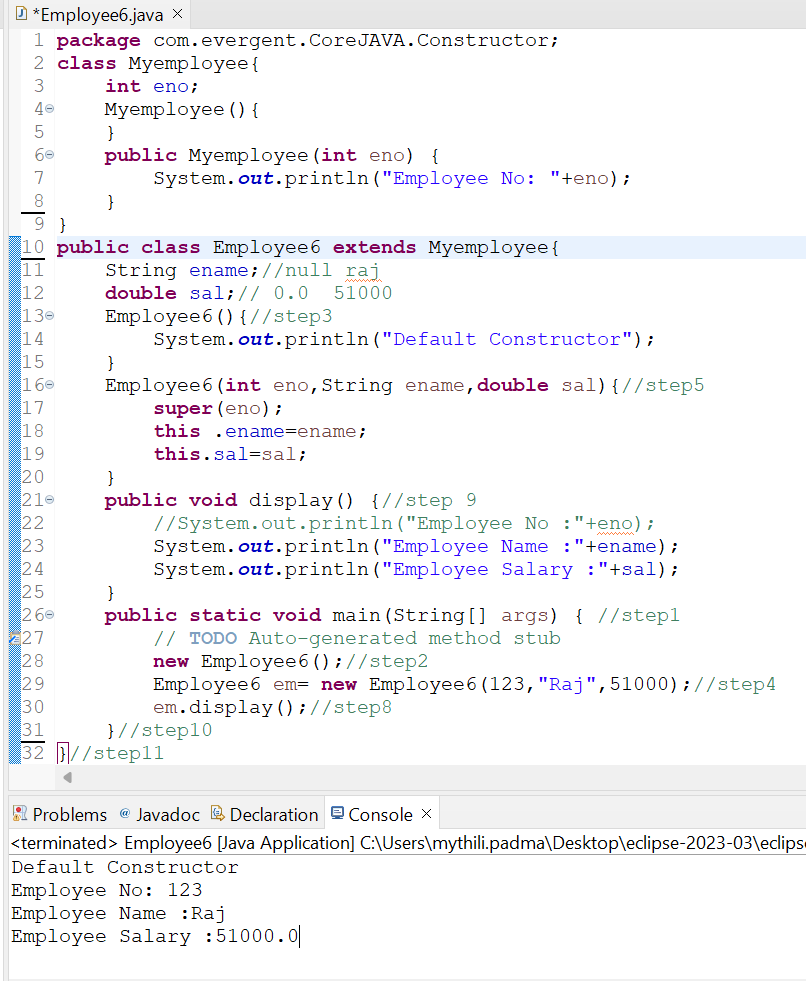
**Program 1: Program 2:**

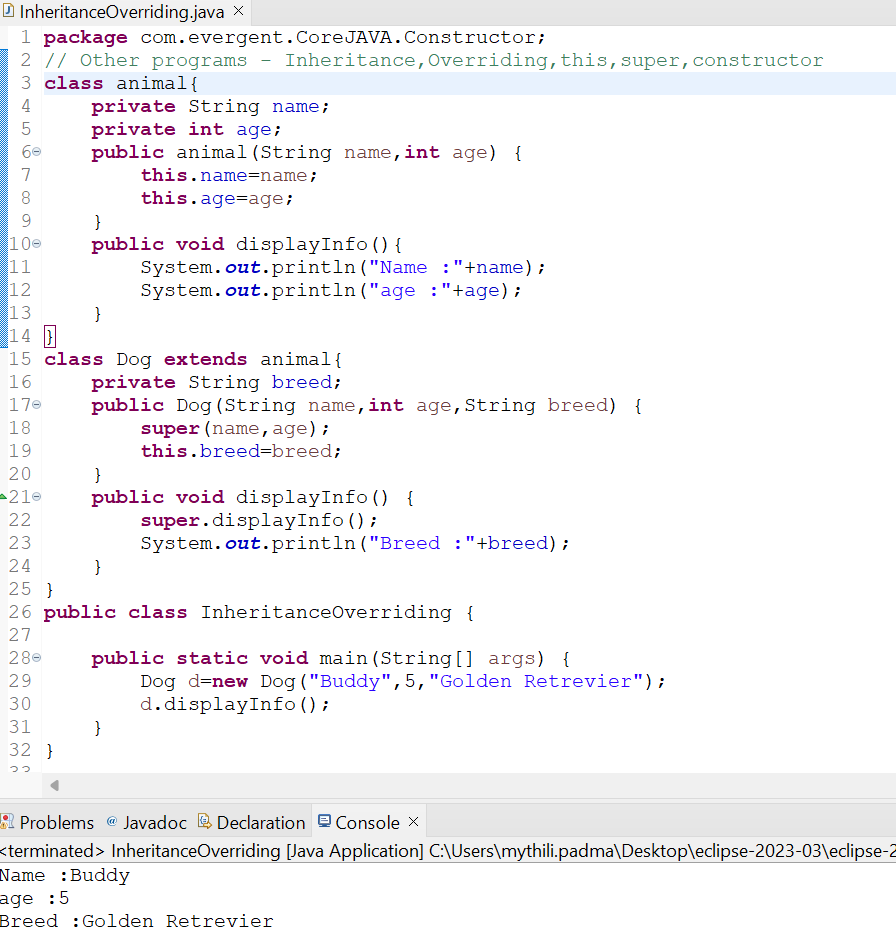
**Program 3: Program 4:**

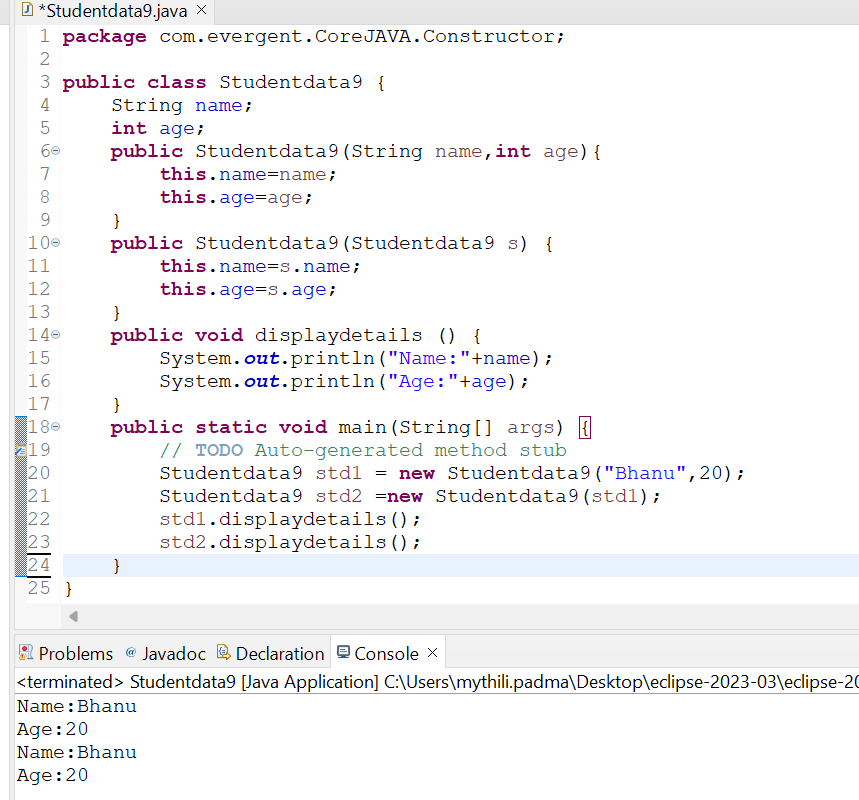
**Program 5:** **Program 6:**

**Program 7:** **Program 8:**

**Program 9:**



**Day -5 (09-08-2024)**

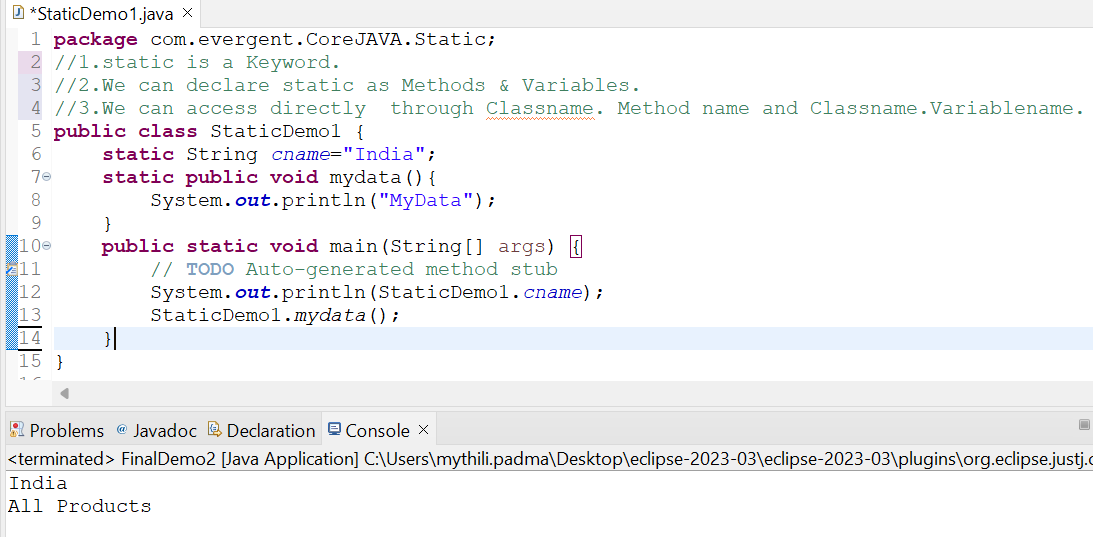
**Static :**

1. static is a Keyword.
2. We can declare static as Methods & Variables.
3. We can access static variables & static methods directly through Classname. Method name and Classname.Variablename.
4. Static methods can access static methods and static variables.
5. Static methods can not access non-static methods and non-static variables.
6. Non-Static methods can access static methods and static variables.
7. Static block ---> whenever class is loaded into JVM at that time static block is initialzed.
8. If Static variable is modified it will reflect globally

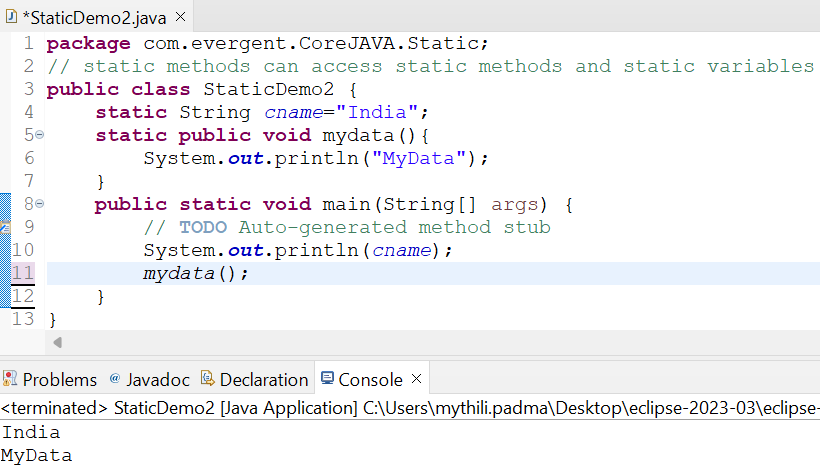
**Final :**

1. final is a Keyword.
2. We can declare final as Variables, Methods & Class.
3. Final variables can not be modify.
4. Final methods can not be Override.
5. Final Class can not be extends (Inherited).

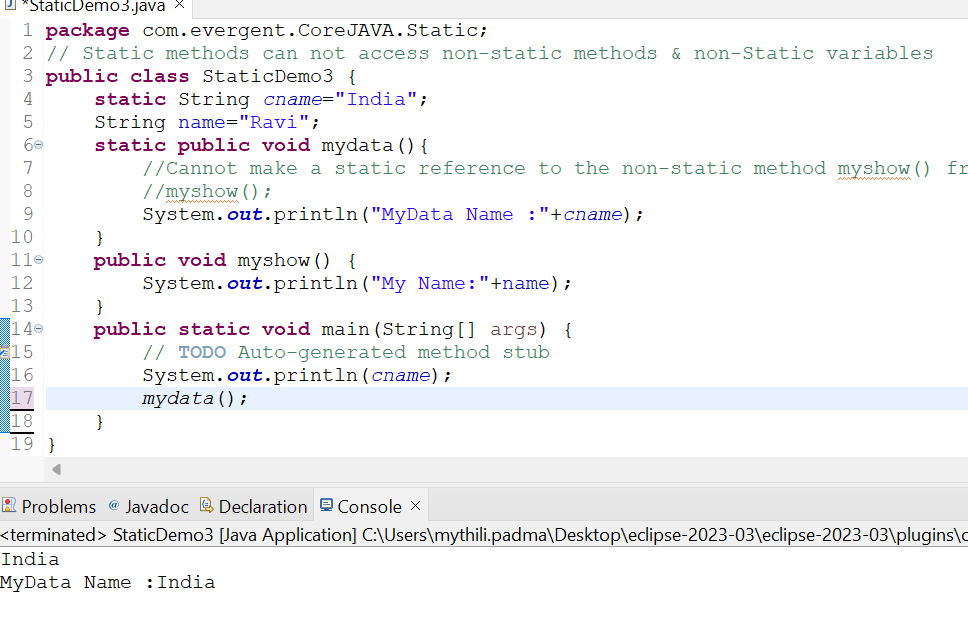
**Program 1:**



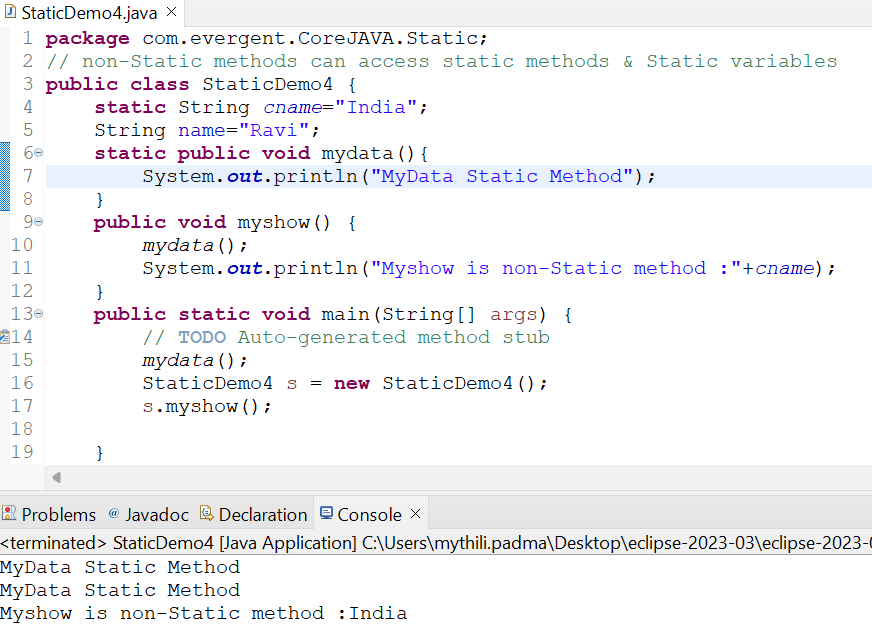
**Program 2:**



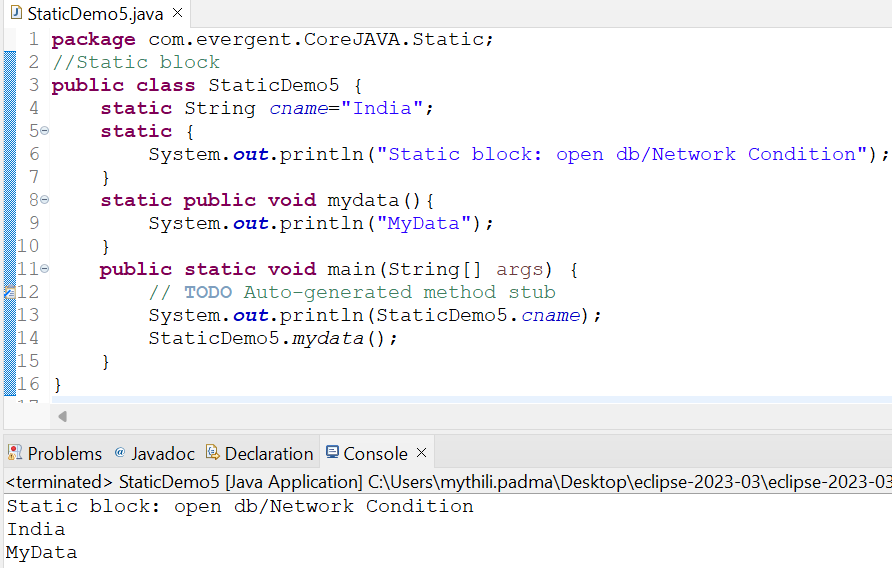
**Program 3:**



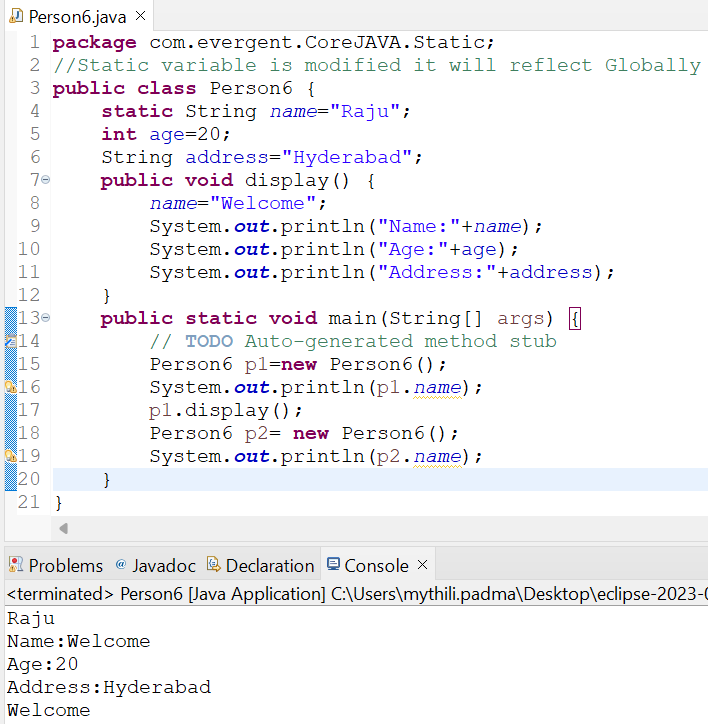
**Program 4:**



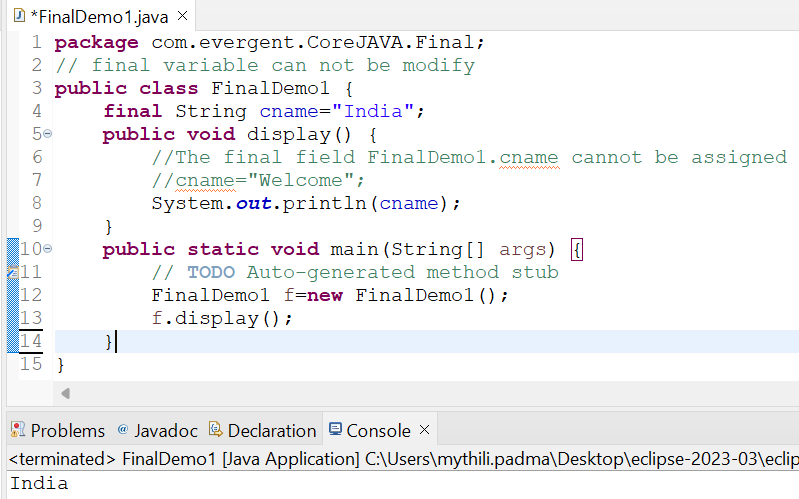
**Program 5:**



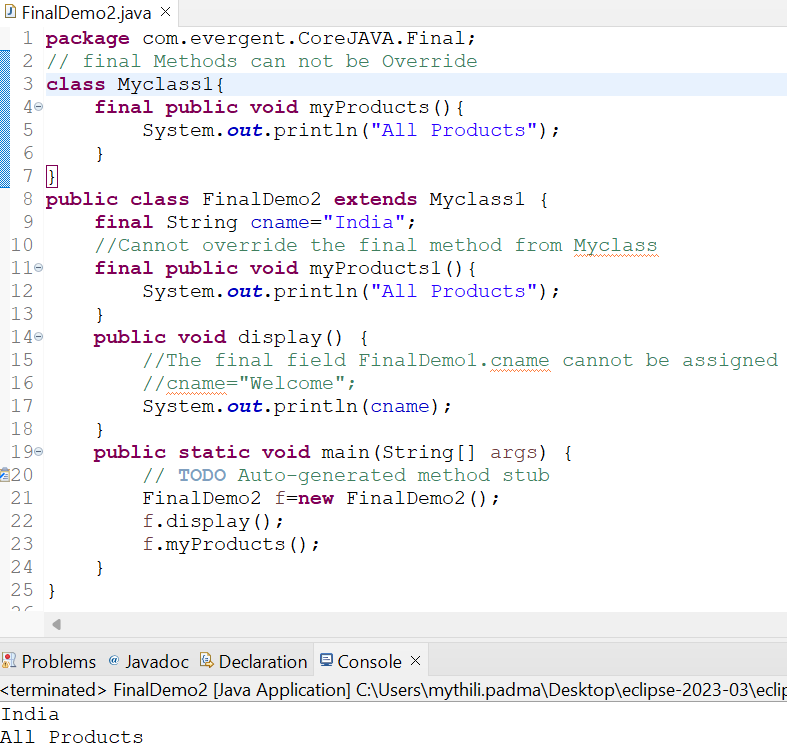
**Program6:**



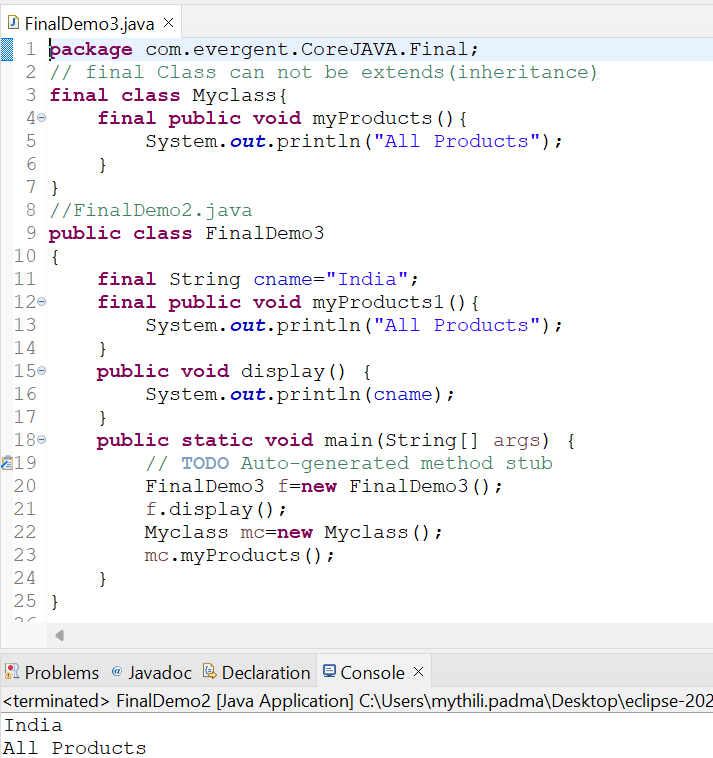
**Program 7:(final)**



**Program 8:**



**Program 9:**



**Day -6 (12-08-2024)**

**String :**

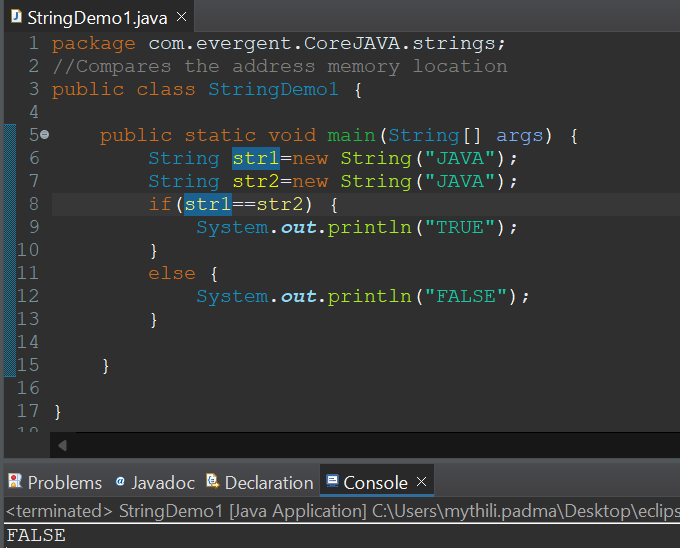
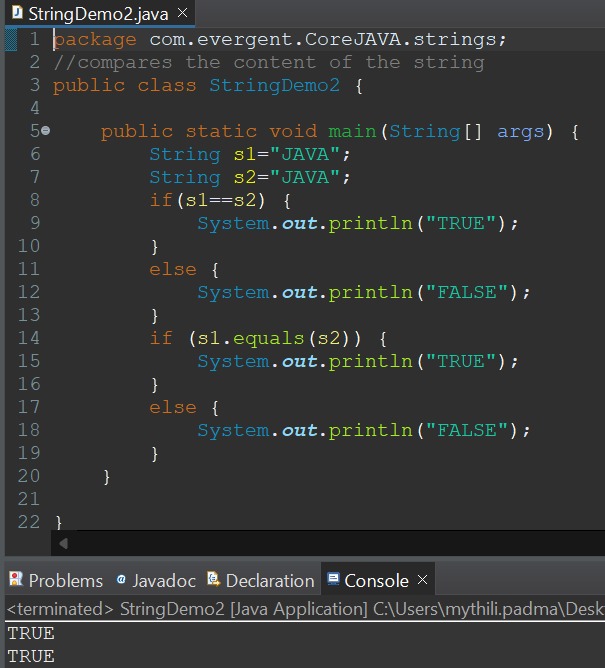
* 1. **String class**  
      -String is Final Class  
      -Strings are immutable  
      -Why strings are immutable?  
      Once we declare any string object, it is constant. If we are trying to modify an existing string it will create another memory location, the existing object is eligible for Garbage Collection.  
      -String class having methods  
      -All string class methods are non-synchronized  
     **1.2 String Buffer**  
      -String is Final Class  
      -Strings are mutable  
      -String class having methods  
      -All string class methods are synchronized

-It is not recommended to use in development but still in API.

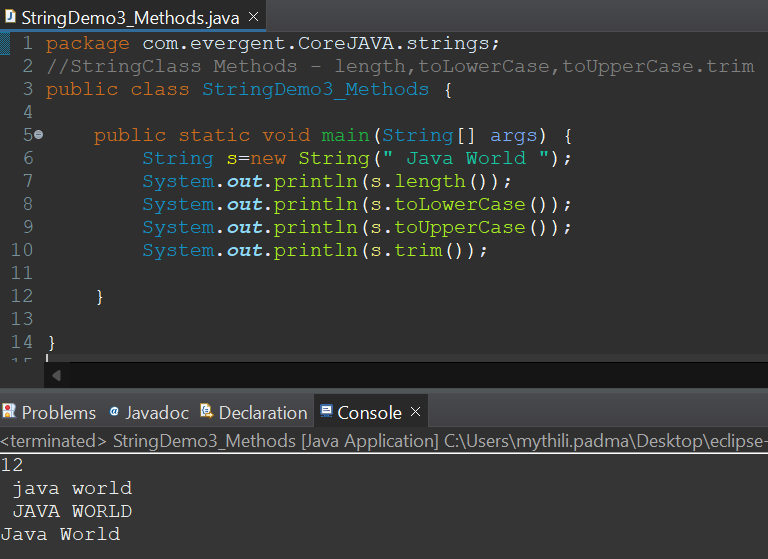
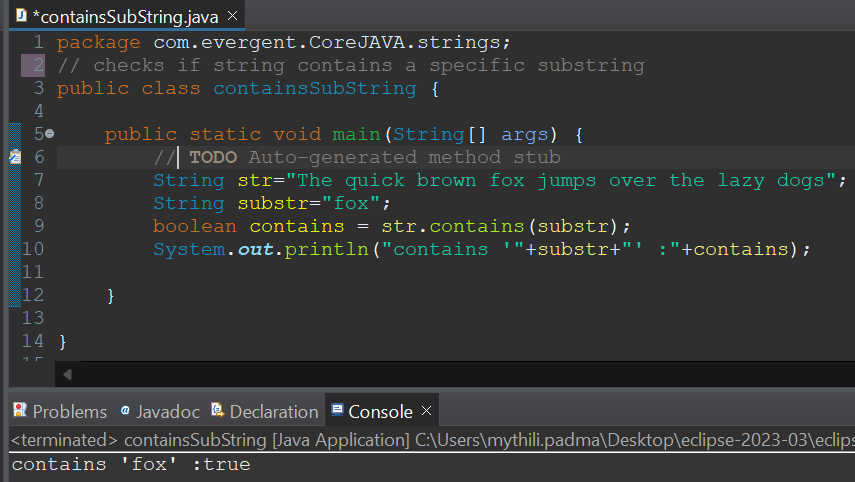
-Methods --> append(),insert(),delete(),replace(),reversed(),capacity(),length()  
**1.3 String Builder** -String is Final Class  
 -Strings are mutable  
 -String class having methods  
 -All string class methods are non-synchronized

-Methods --> append(),insert(),delete(),replace(),reversed()

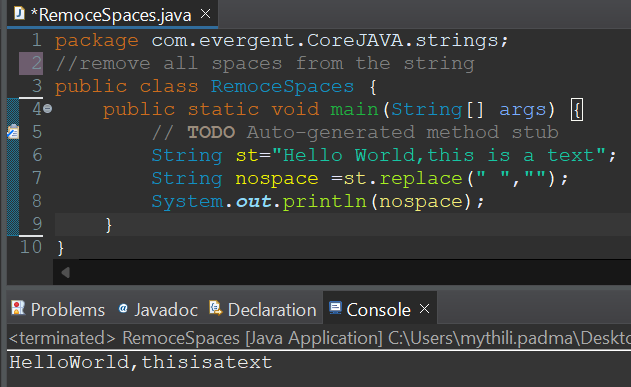
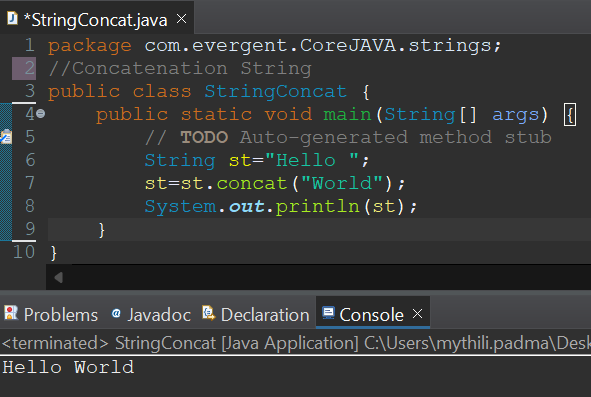
**Program 1: Program 2:**

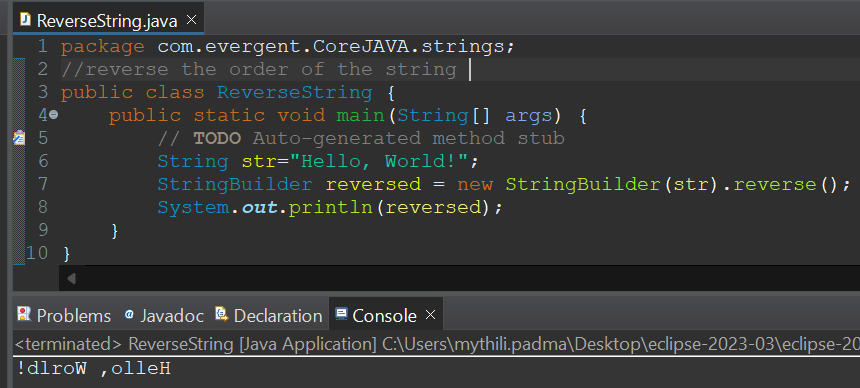
**Program 3: Program 4:**

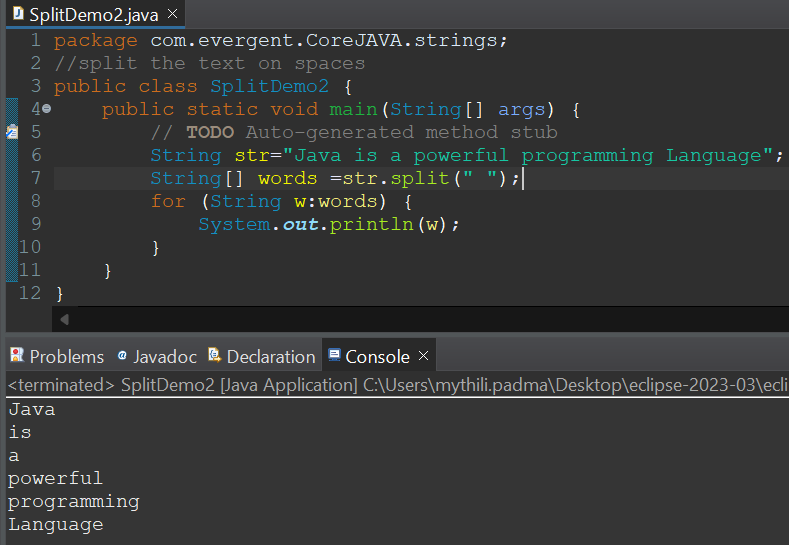
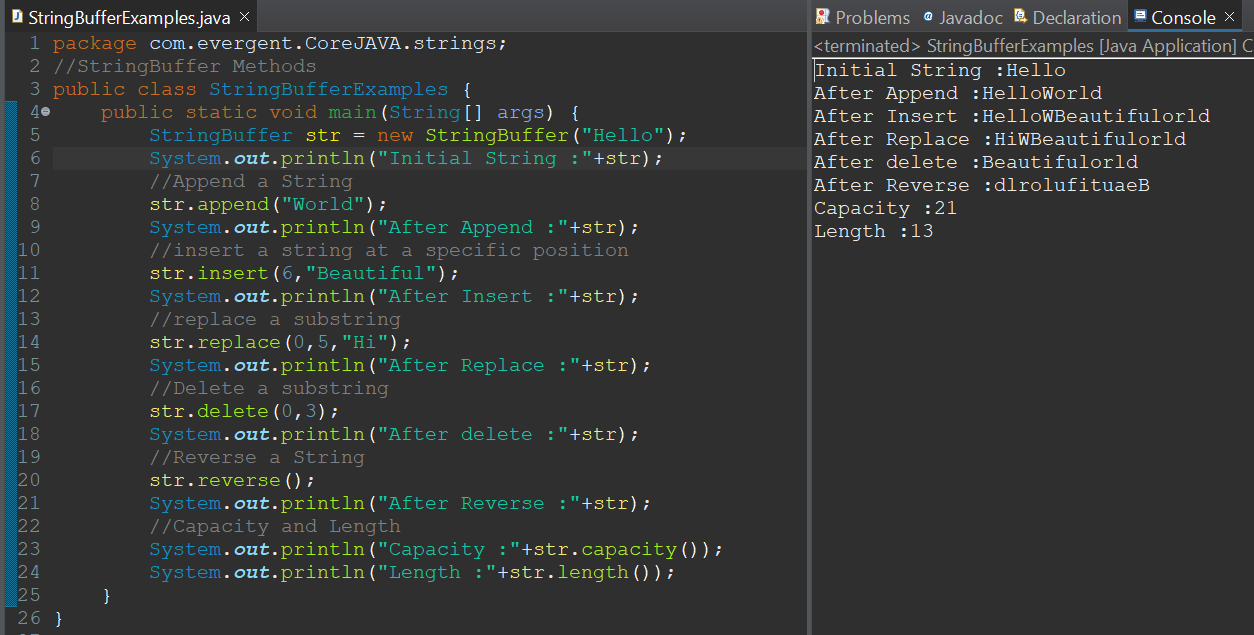
**Program 5: Program 6:**

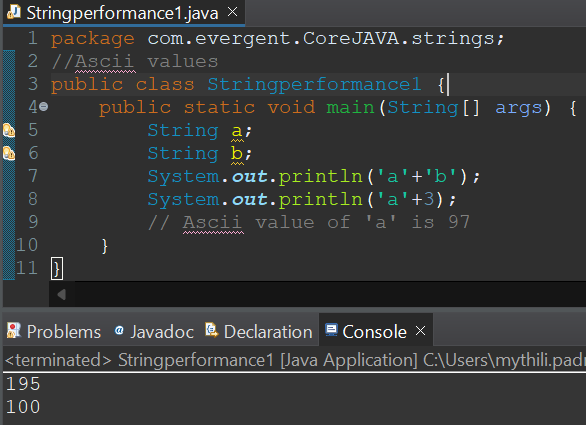
**Program 7: Program 8:**

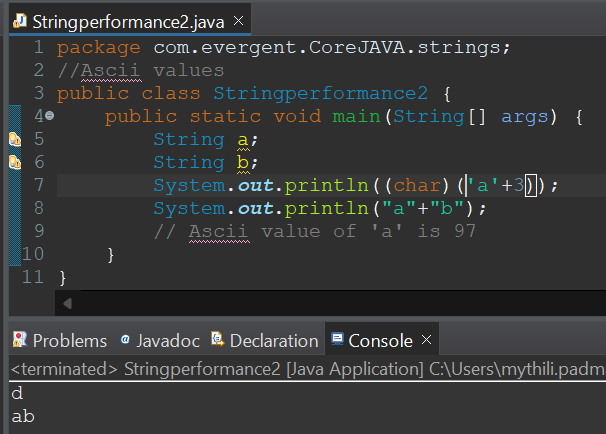
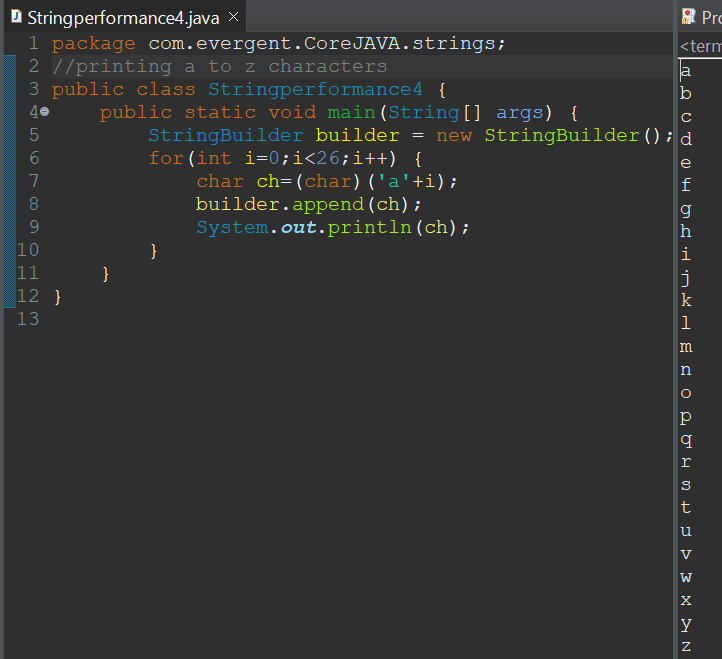
**Program 9: Program10:**

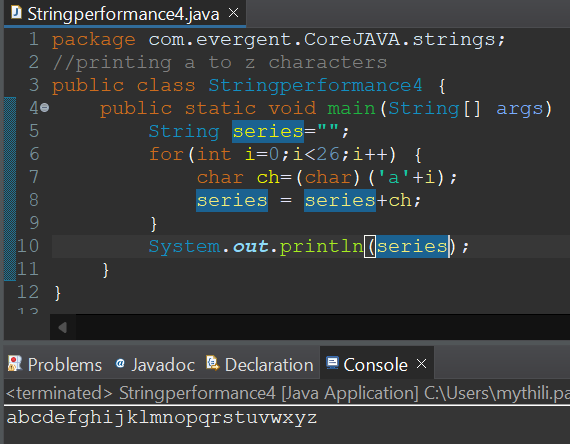
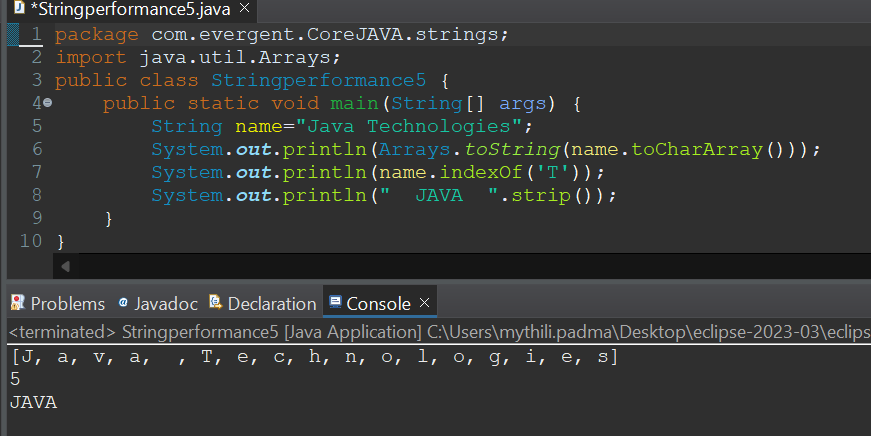
**Program 11: Program 12:**

**Program 13: Program 14:**

**Program 15: Program 16:**

**Day-7 (13-08-2024)**

* **Immutable**

1.We can create our own immutable class?

a.We can declare class as final.

b.The class is declared as final so that it cannot be sub classed..

c.Private final fields:

i.The fields name and age are private and final constructor.

d.Constructor:

ii.The constructor initializes the final fields when the person object is created.

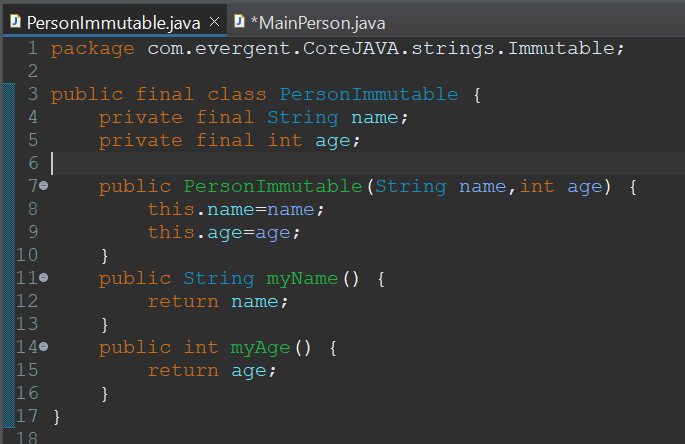
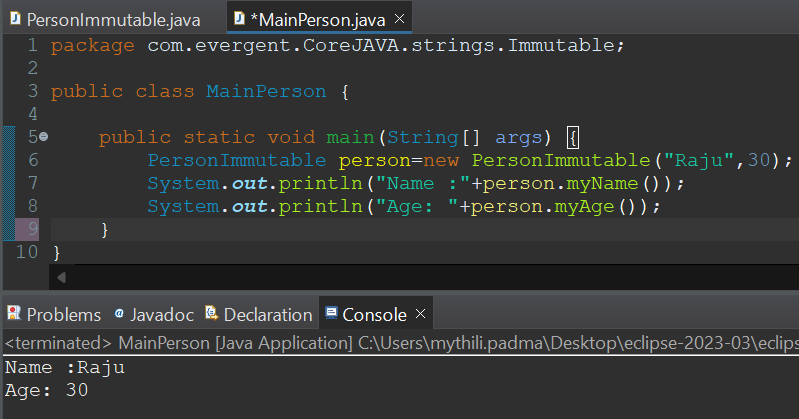
* **Object Class Methods :**

1. toString()
2. hashCode()

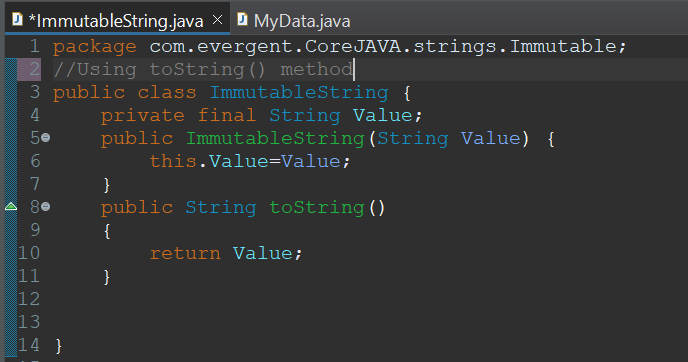
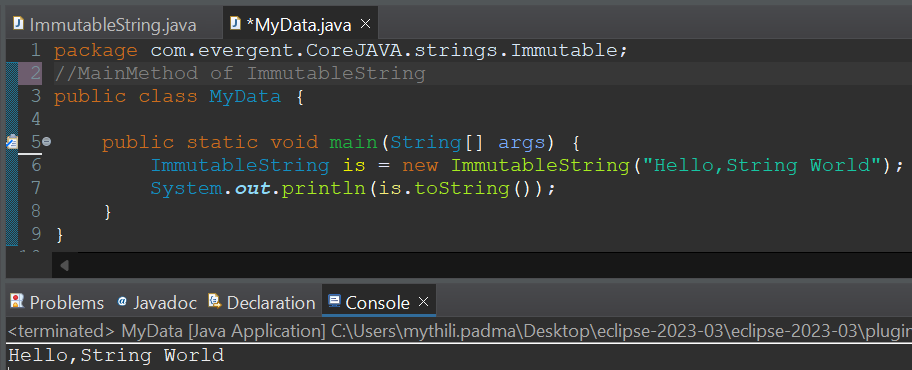
* I**nterfaces:**

1. Interface is a keyword.
2. We can declare methods signatures only but not implementations.
3. By default all interface methods are abstract.
4. If any class implements interface that class should override all interface methods otherwise the class will be showing compile time error.
5. We cannot create objects to interface, but we can create references to interface.
6. We can declare variables inside the interface all are public static final.
7. Java will support Multiple inheritance through interface.
8. One class can implements more than one interface.
9. One interface can extend other interface.

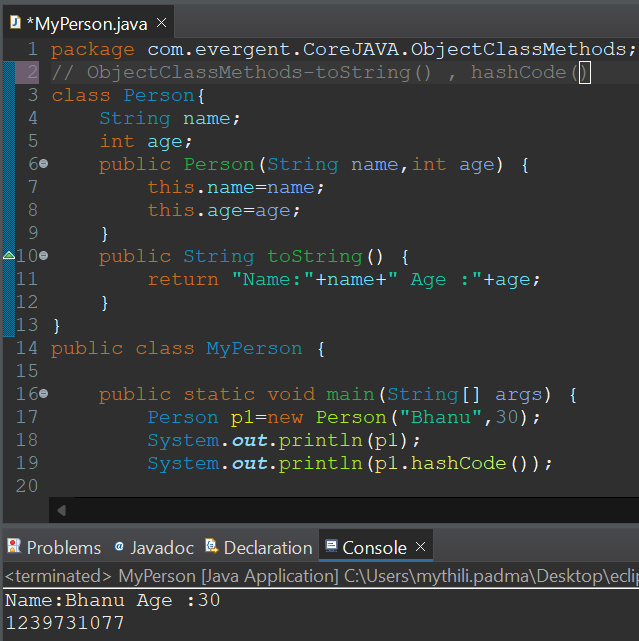
**Program 1:**

** ---> **

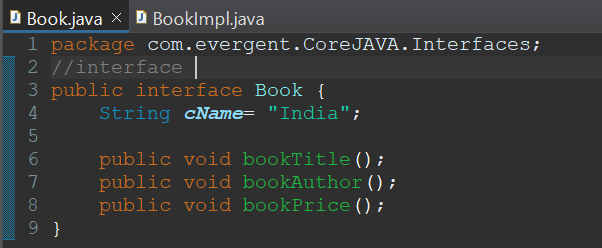
**Program 2:**

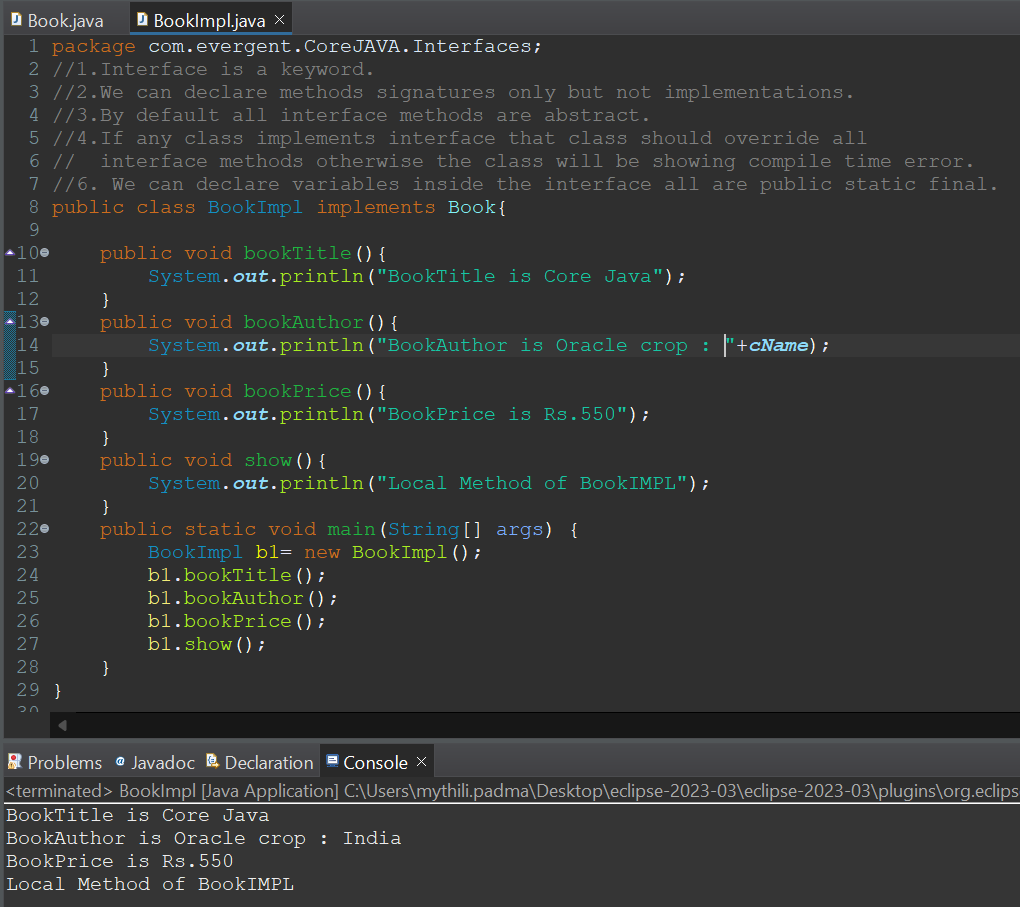
** ---> **

**Program 3:**

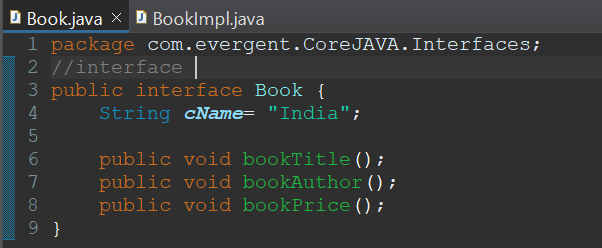
****

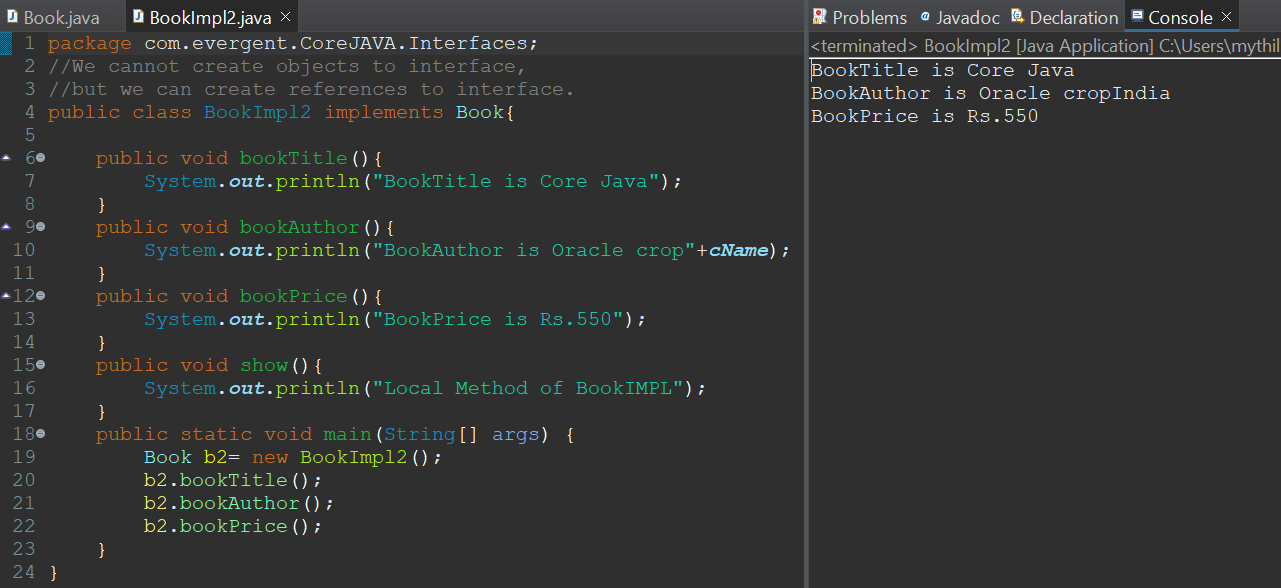
**Program 4:**

****

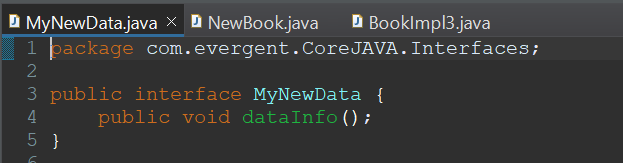
****

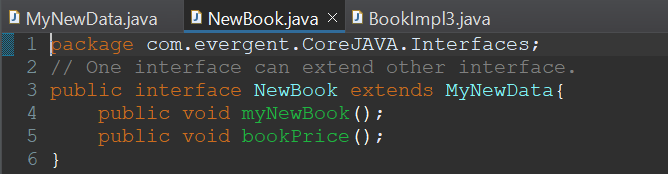
**Program 5:**

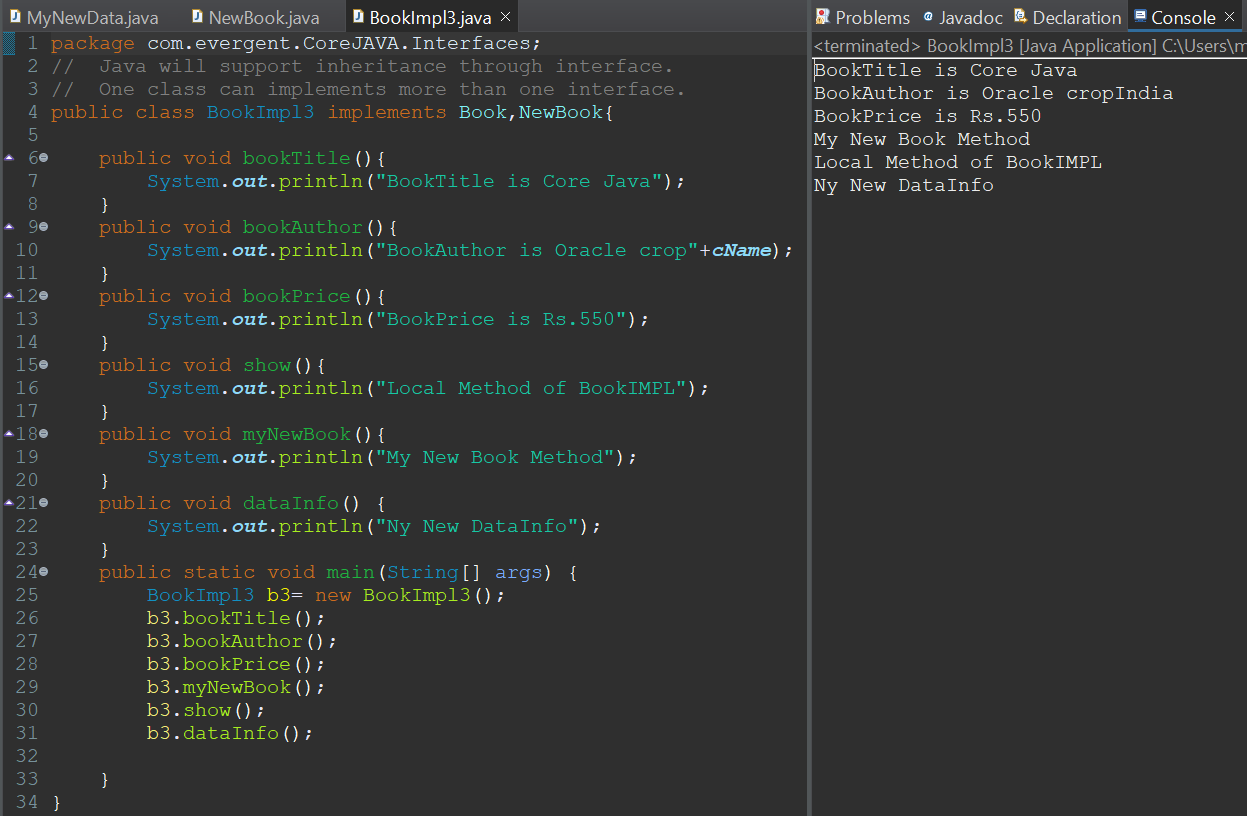
****



**Program 6:**

****

****

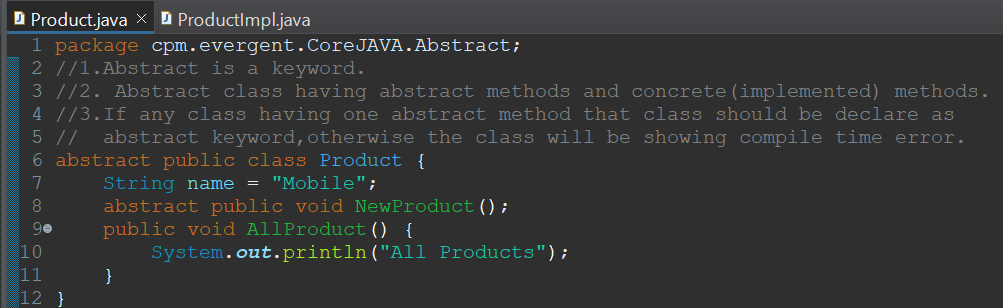


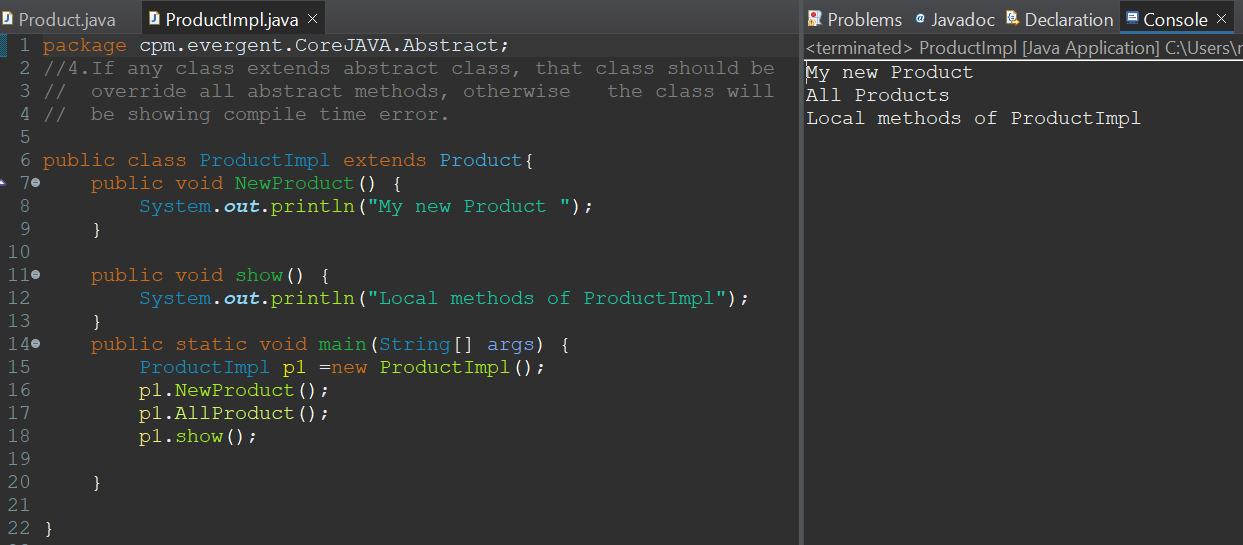
**Day-8 (14-08-2024)**

* **Abstract Class**

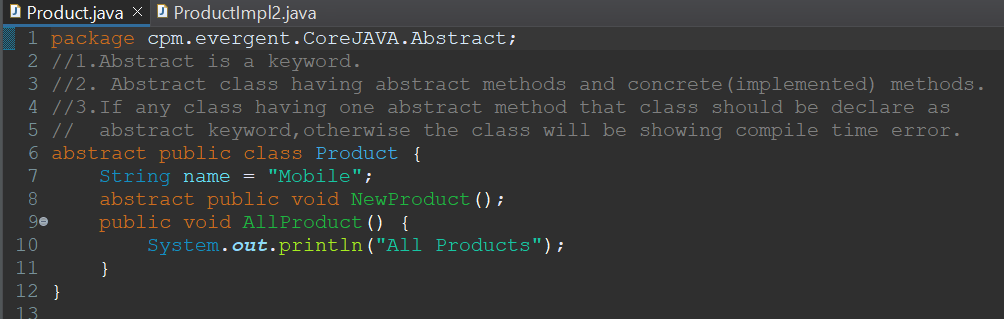
1. Abstract is a keyword.
2. Abstract class having abstract methods and concrete(implemented) methods.
3. If any class having one abstract method that class should be declare as abstract keyword, otherwise the class will be showing compile time error.
4. If any class extends abstract class, that class should be override all abstract all methods, otherwise the class will be showing compile time error.
5. We cannot create object to abstract class but we can create reference to abstract class.
6. We can declare abstract class with zero abstract methods.
7. We Can Create Constructor to Abstract Class.
8. We Can Access abstract class constructor through Subclass.

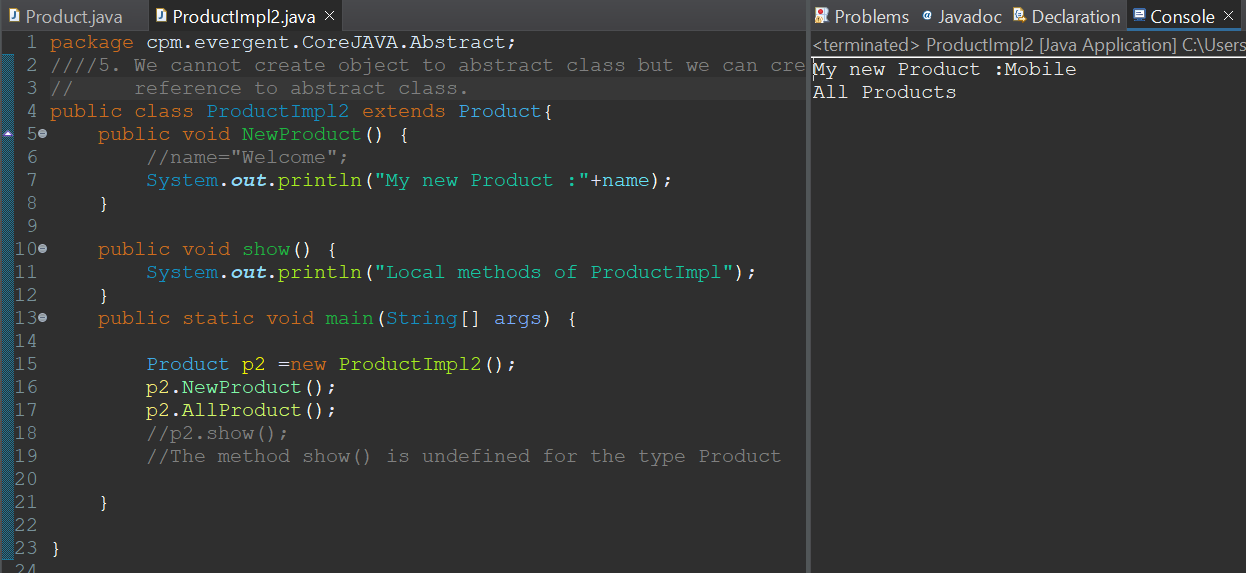
**Program 1:**

****

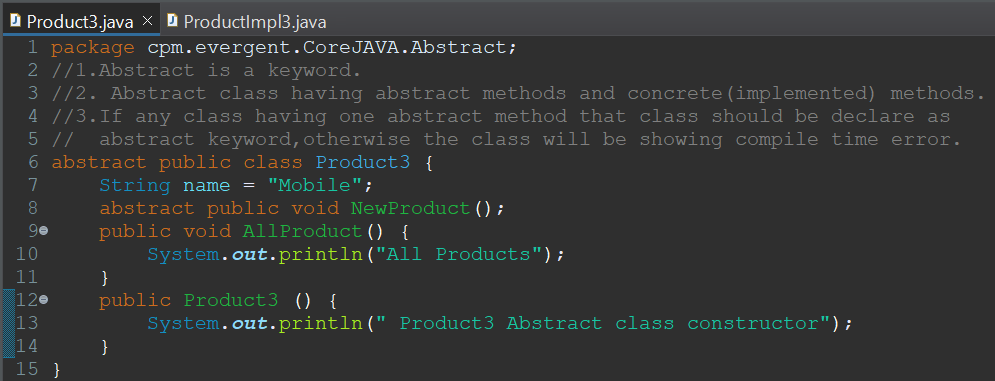


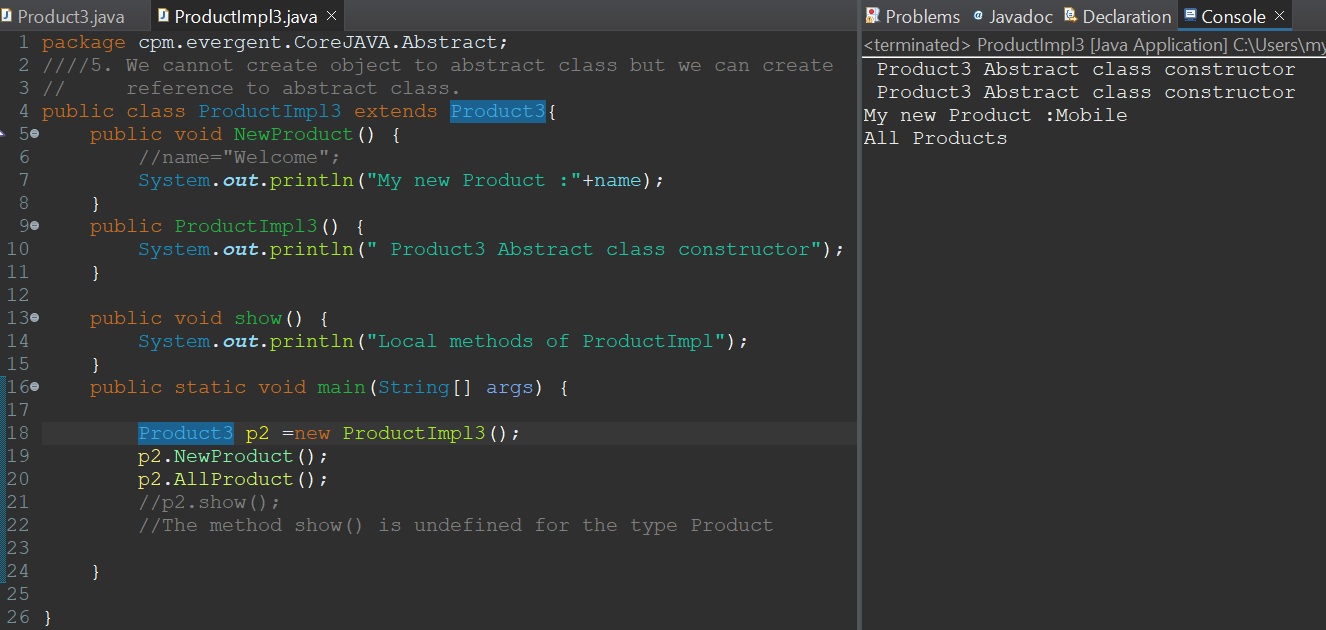
**Program 2:**

****



**Program 3:**

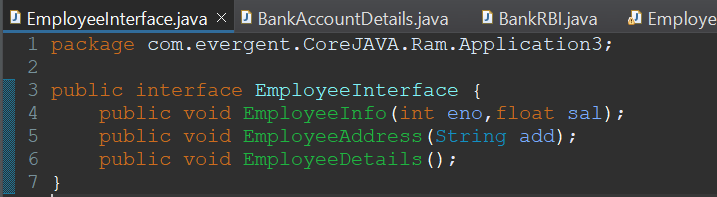
****

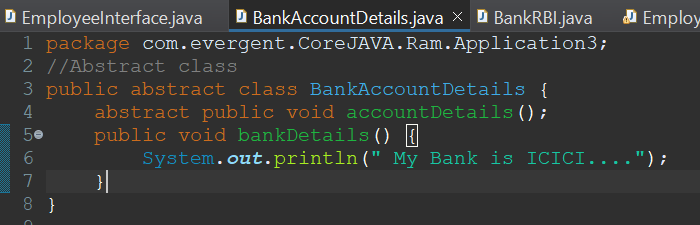
****

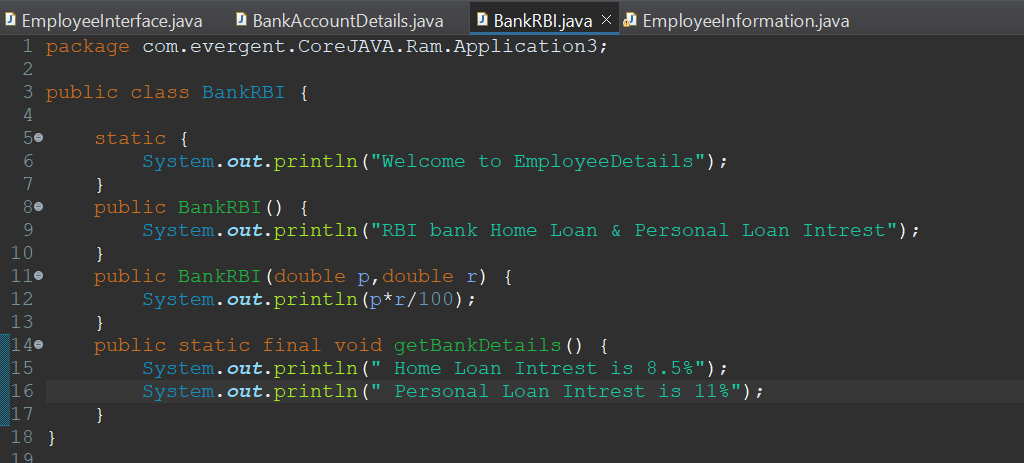
**Day-9 (17-08-2024)**

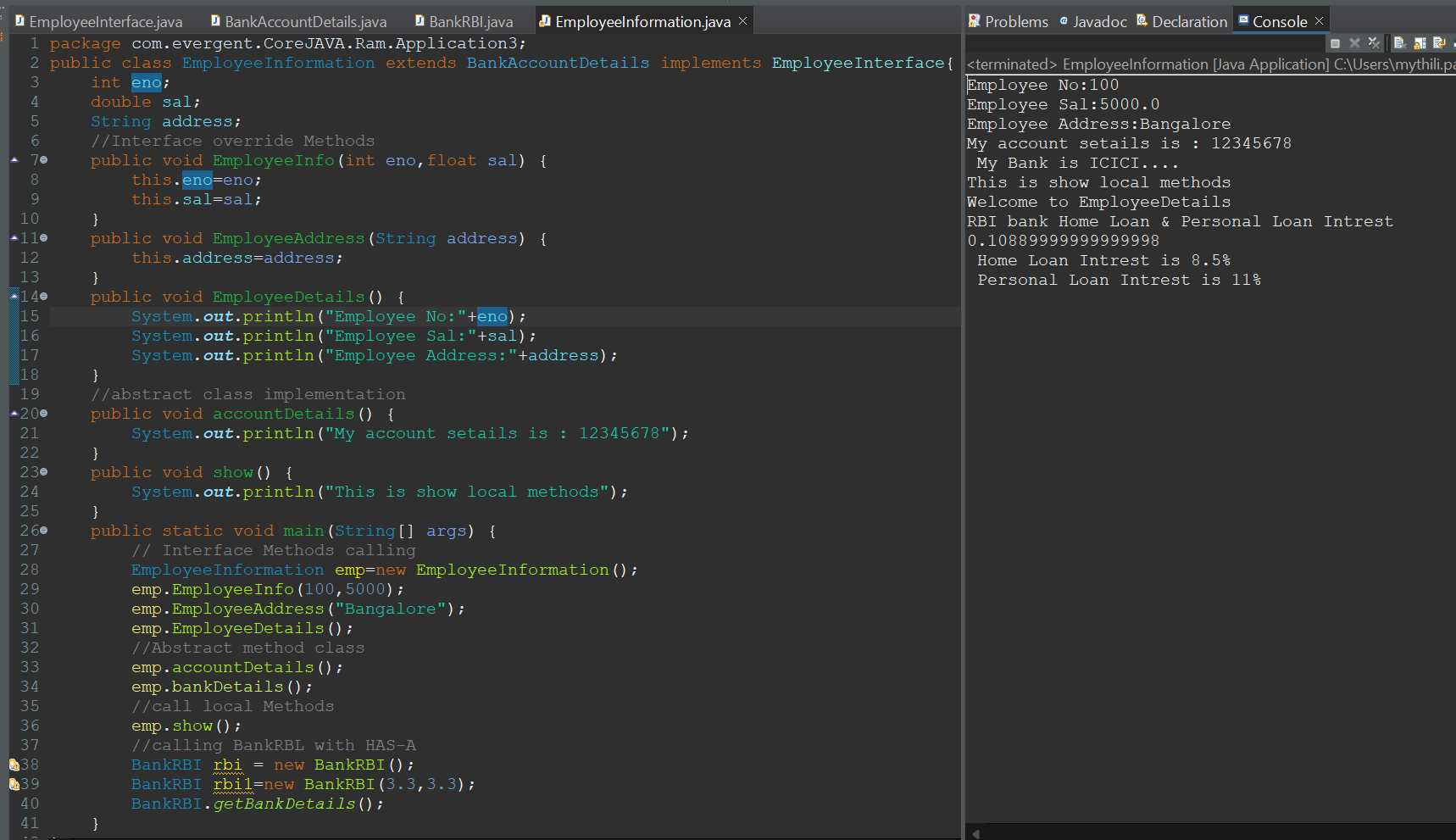
* Conducted the coding test
* Explained about Application & about Interview Questions

**Application 3 Program:**

****

****

****

****

**Day-10 (19-08-2024)**

**1.Revision of completed topics: oops, constructors, static, final, strings, interfaces, abstract classes.**

**2. Exception Handling:**

a)Exception Handling is a Mechanism.

b)Exceptions are inbuilt Mechanism of JAVA.

c)All Exceptions are executed during abnormal conditions only.

d)Normal flow it won’t execute any exceptions.

e)Once any exceptions are occurring in java code, then remaining lines of code is unreachable.

f)Java.lang.Throwable is super class for exception and errors.

g)There are two types of Exceptions in JAVA:

a.Checked Exceptions

b.Unchecked Exceptions

h)All checked Exceptions are compile time exceptions.

i)All unchecked Exceptions are run time exceptions.

j)There are 5 keywords in Exception Handling:

a.Try

b.Catch()

c.Finally()

d.Throws()

e.Throw

k)Try is for business logic.

l)Catch for handling exceptions.

m)Finally is a block, if exceptions occurs or not finally block will be executed.

n)Throws an exception will be executed method by method

o)Throw is for runtime exceptions and will call predefined exceptions or user defined exceptions.

p)Try followed by either catch block or finally block.

q)We should follow exceptions hierarchical.

r)We can create our own (User Defined) exceptions.

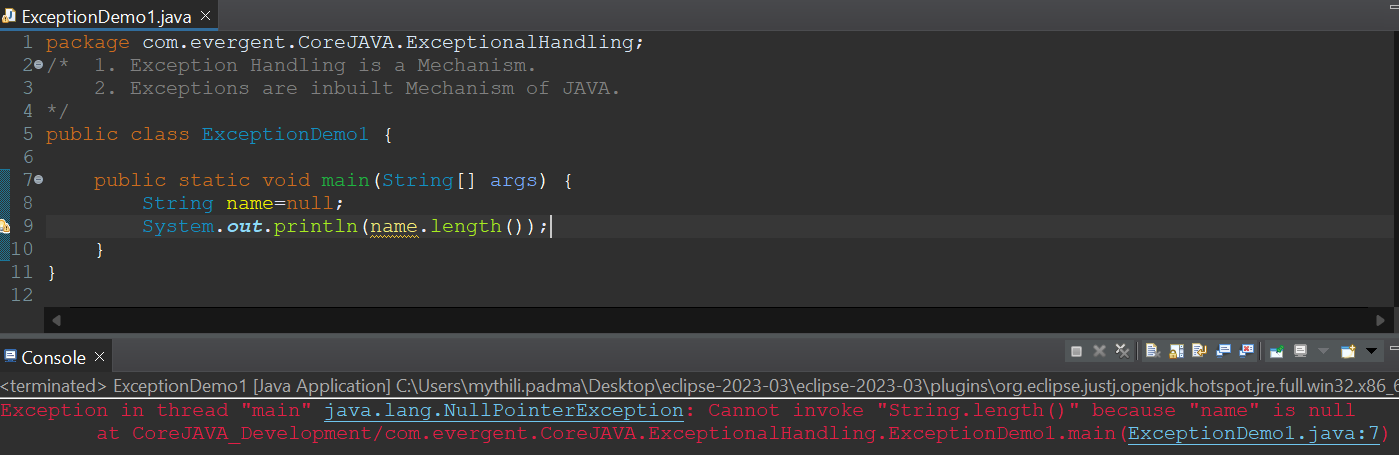
s)Our own exceptions extends exception or runtime exception.

t)All Exceptions classes are in to java.lang package.

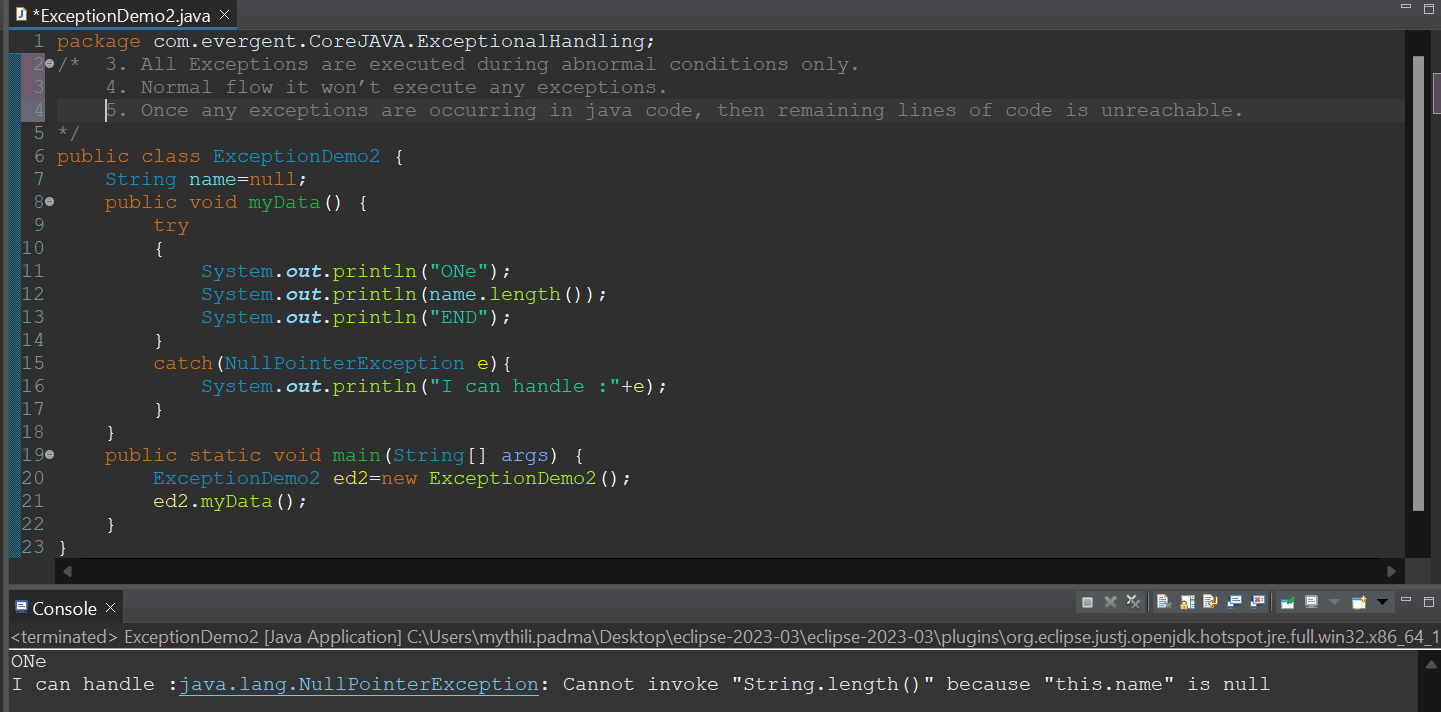
u)If there are 2 exceptions in try, developer should handle 1st exception after that 2nd exception will be handled.

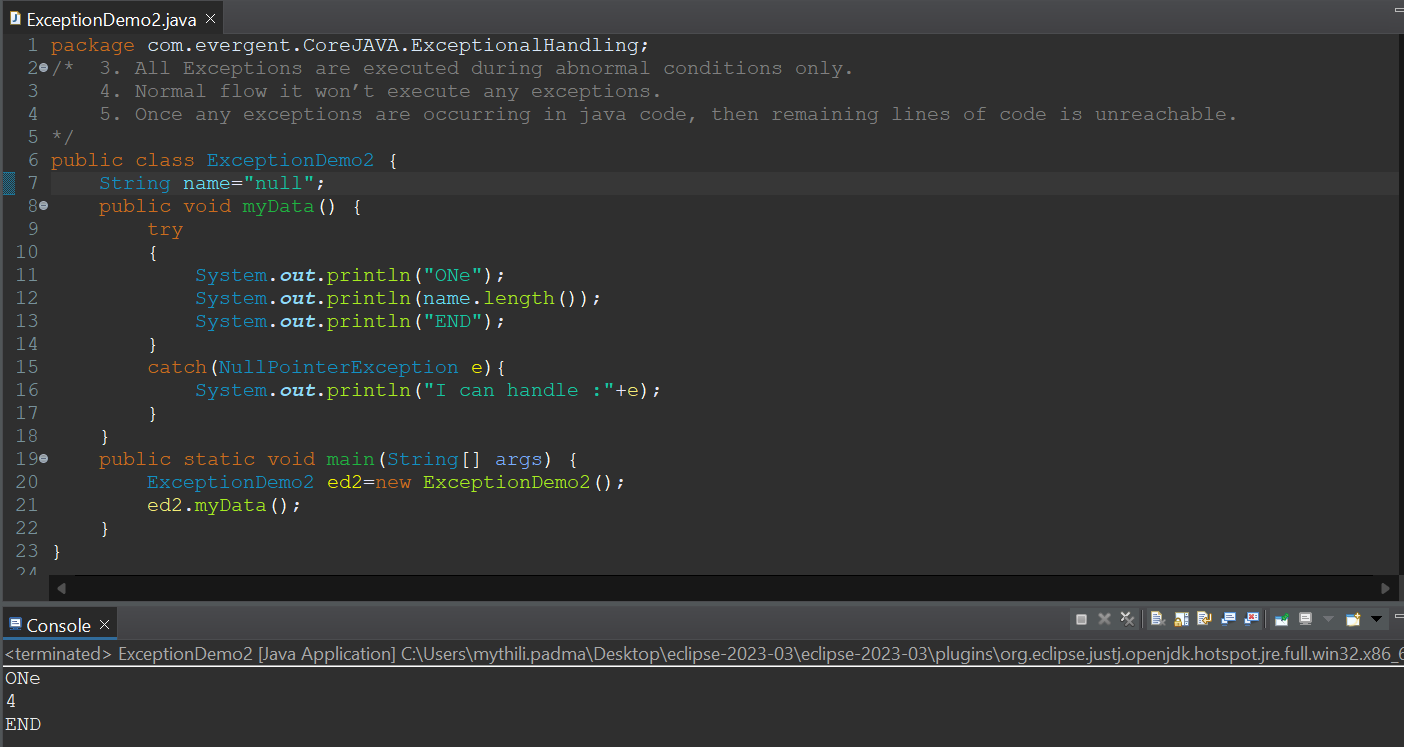
v)Developer can’t handle errors.

**Program 1:**

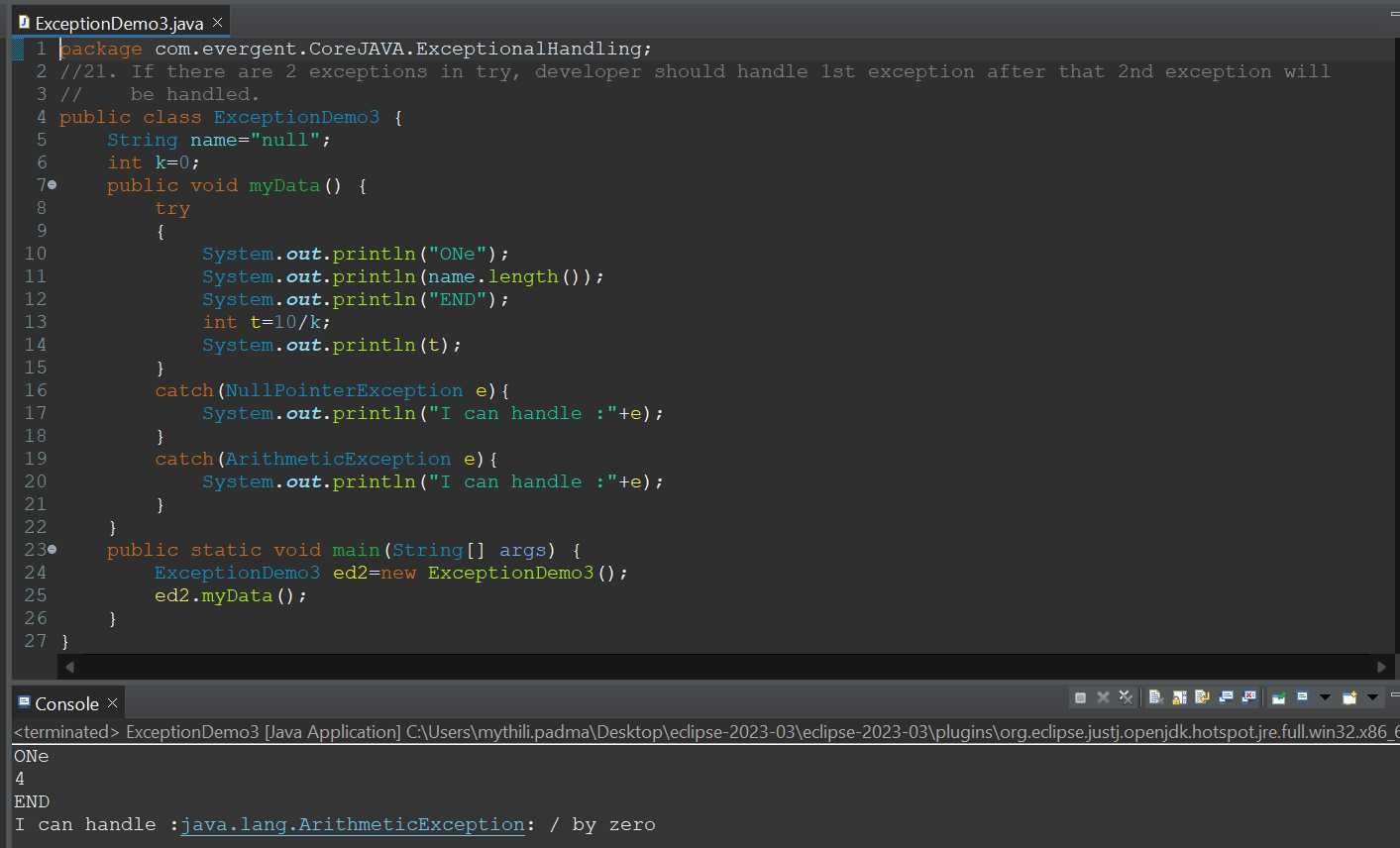
****

**Program 2:**

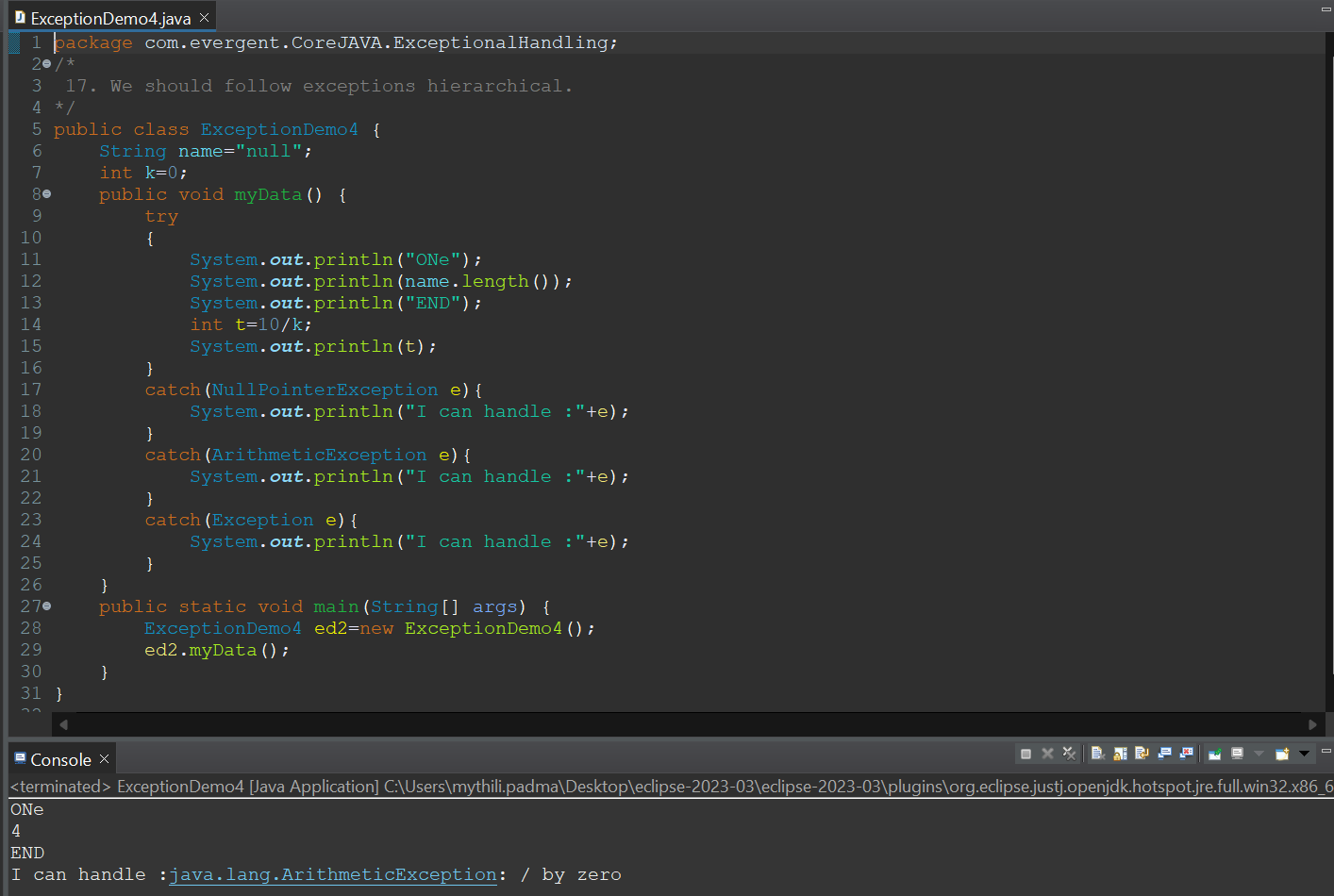
****



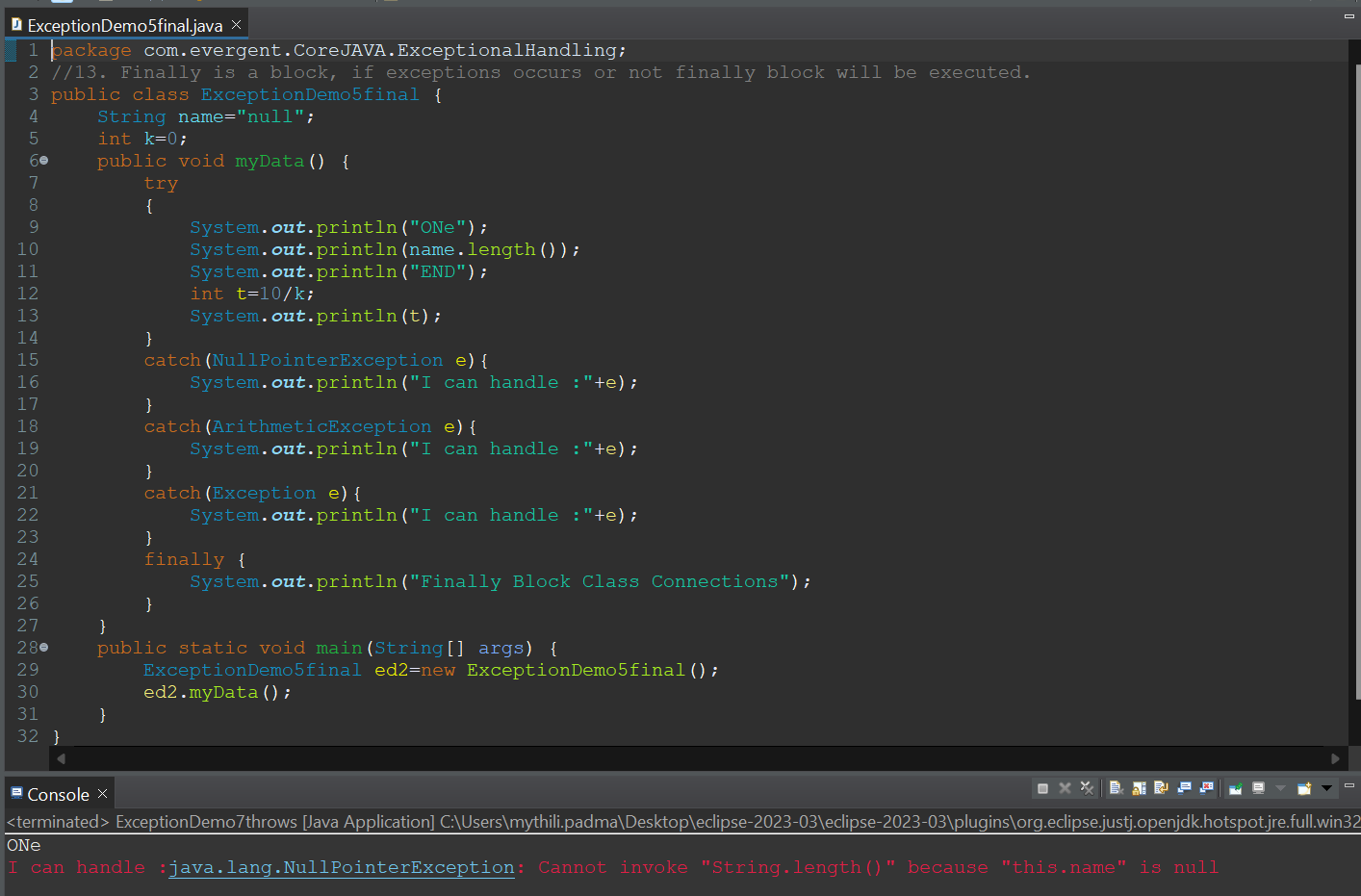
**Program 3:**



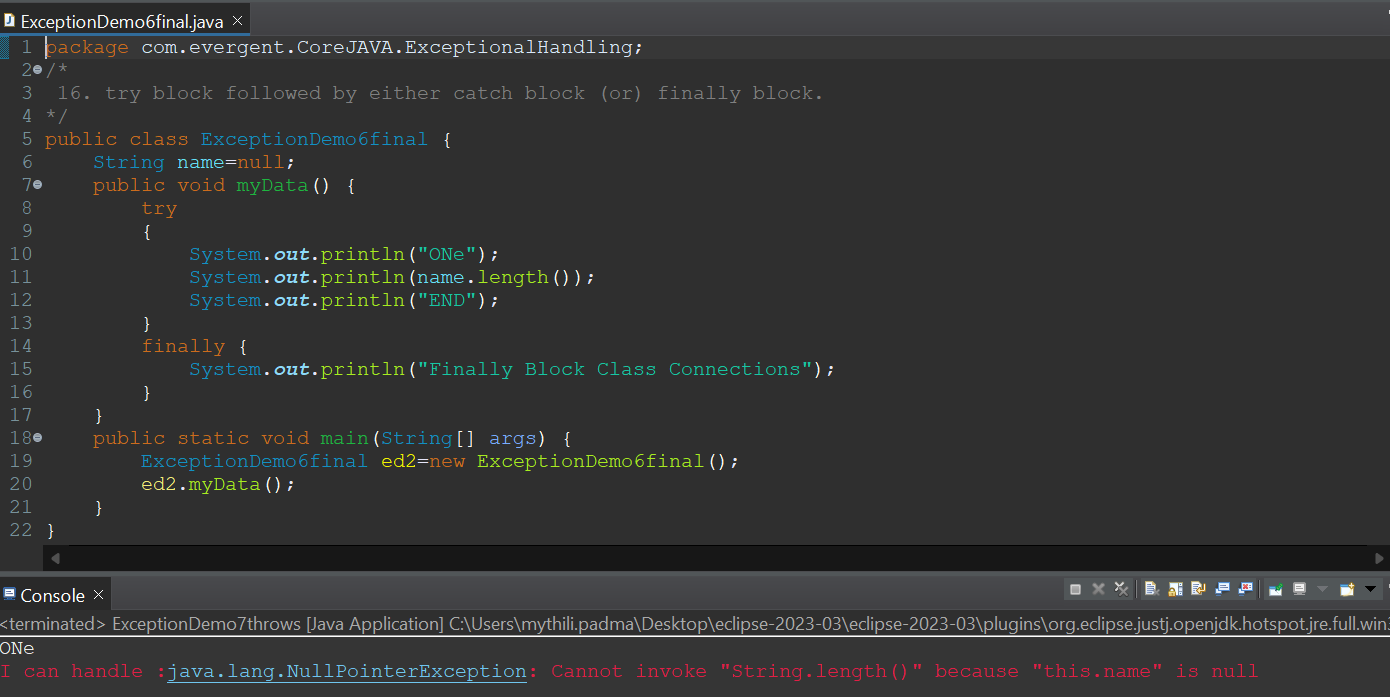
**Program 4:**



**Program 5:**



**Program 6:**

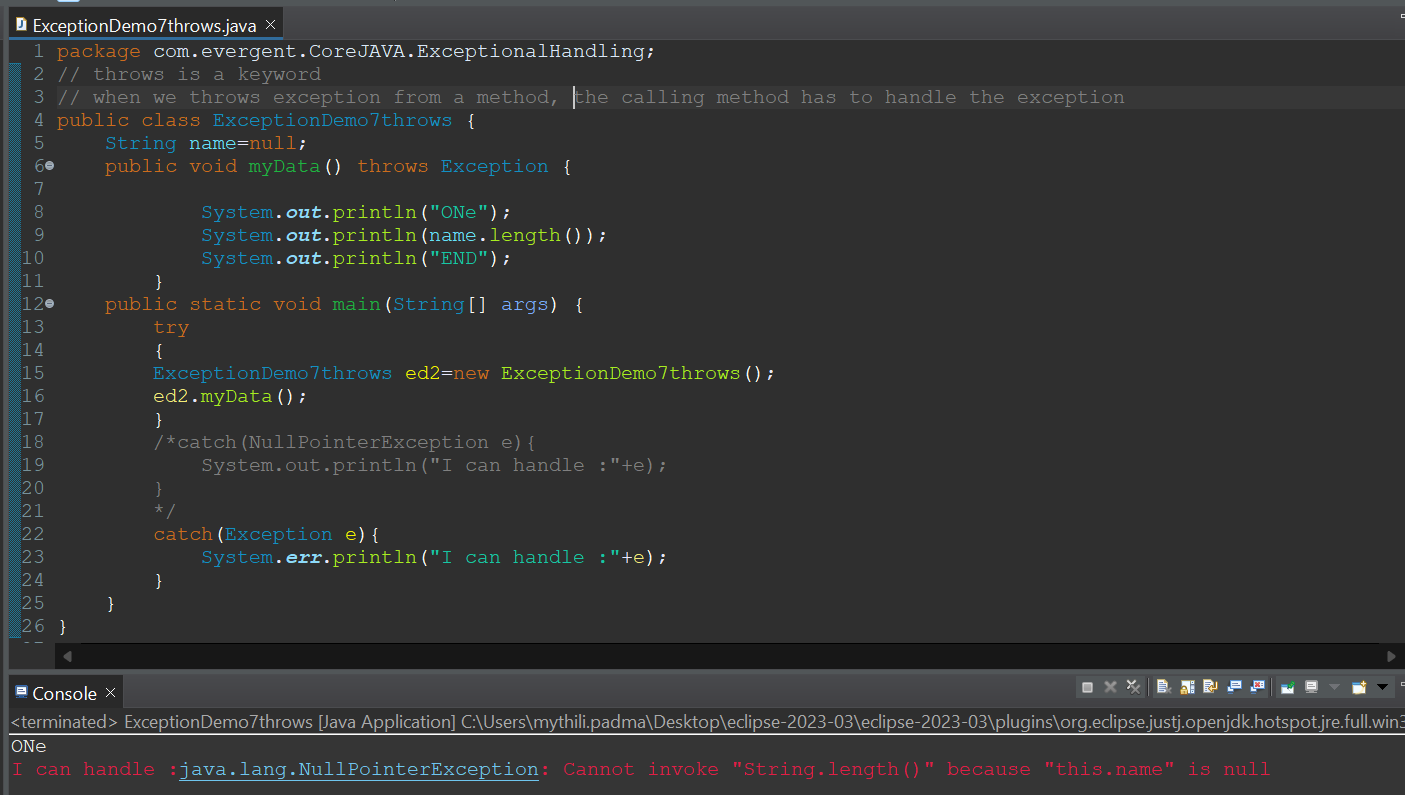


**Day-11 (20-08-2024)**

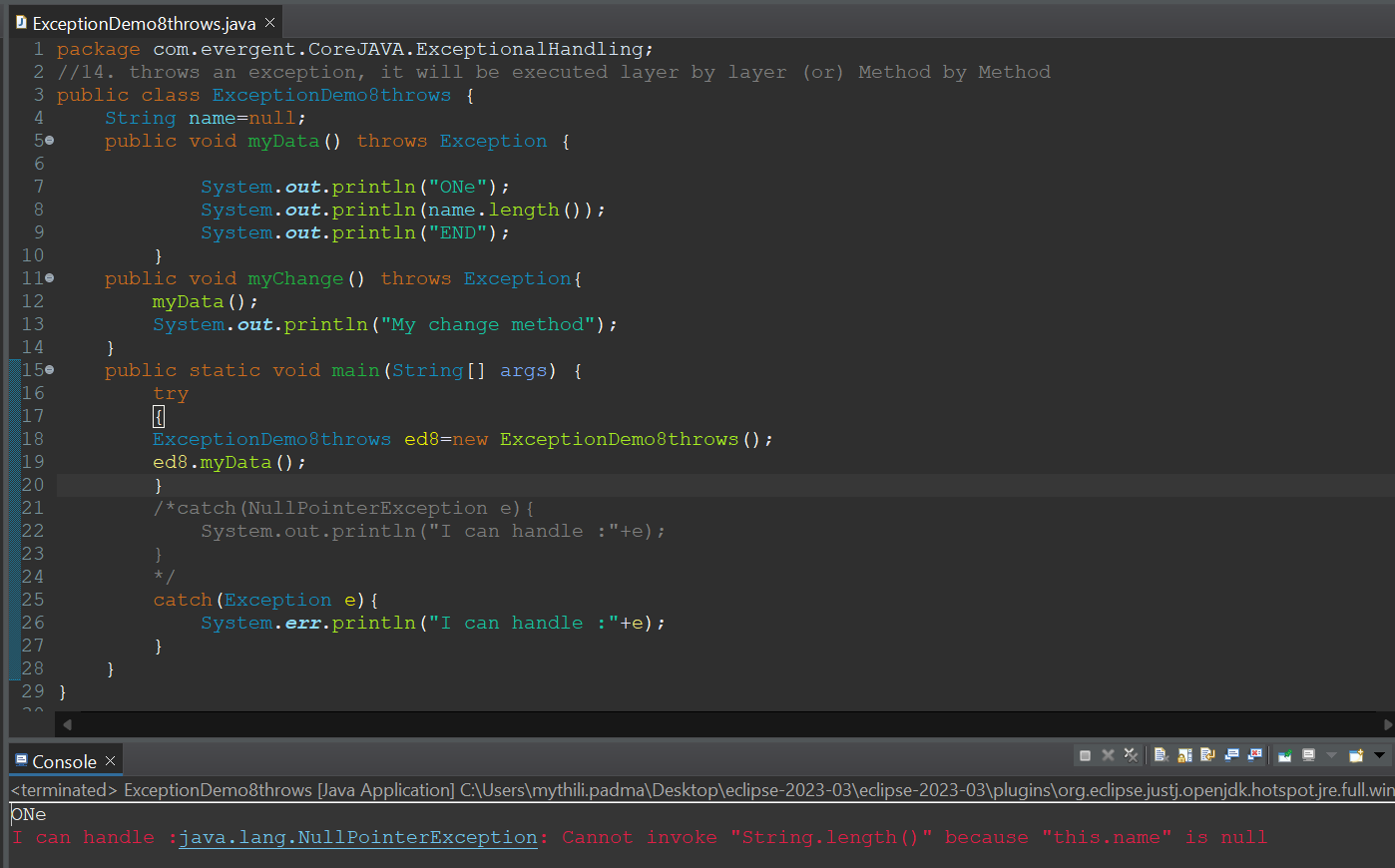
* **Throws**

1. Throws is a keyword.
2. Throws an exception will be executed method by method.
3. When we throws an exception from a method ,the calling method should handle the exception.

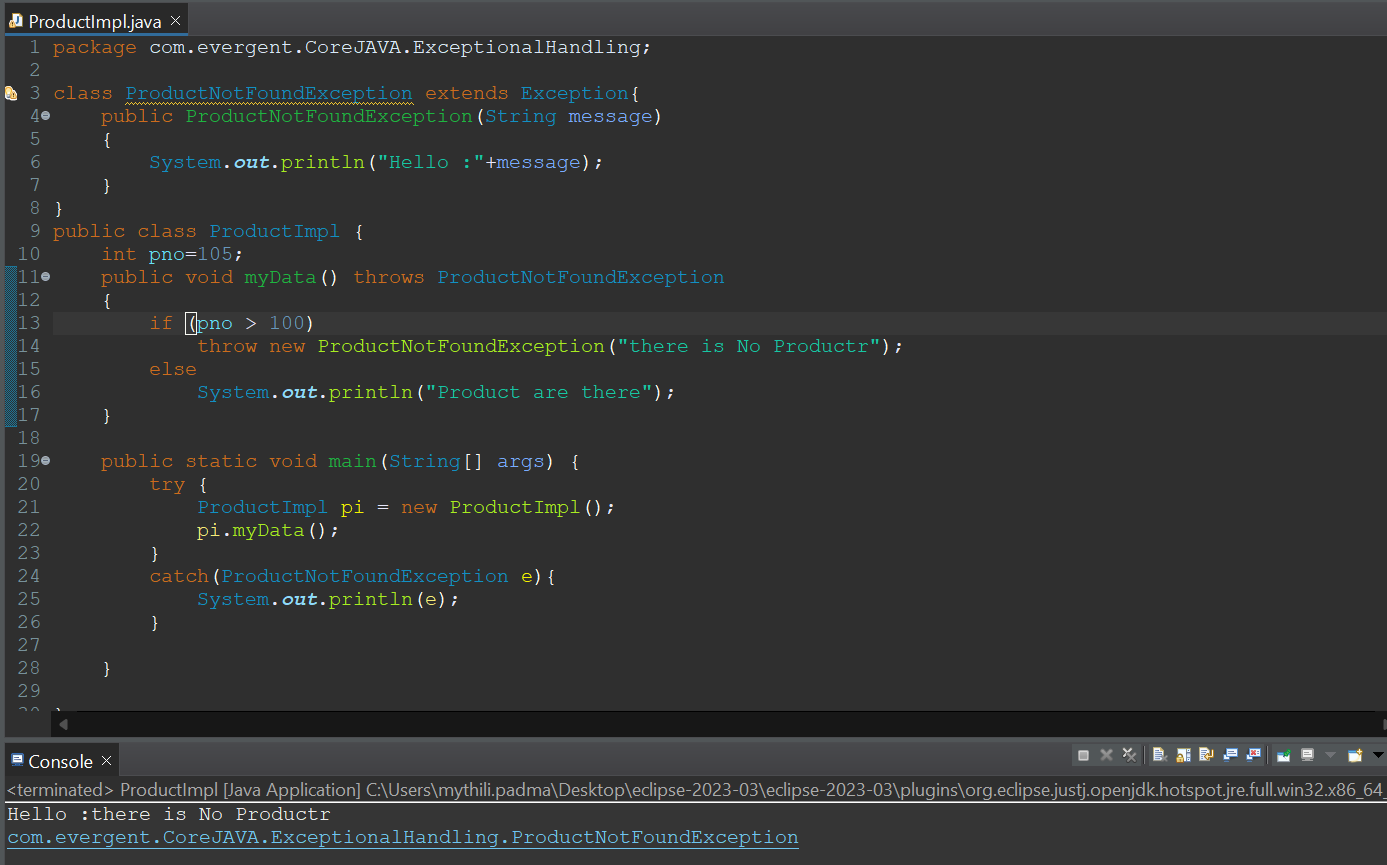
**Program 1:**

****

**Program 2:**



**Program 3:**

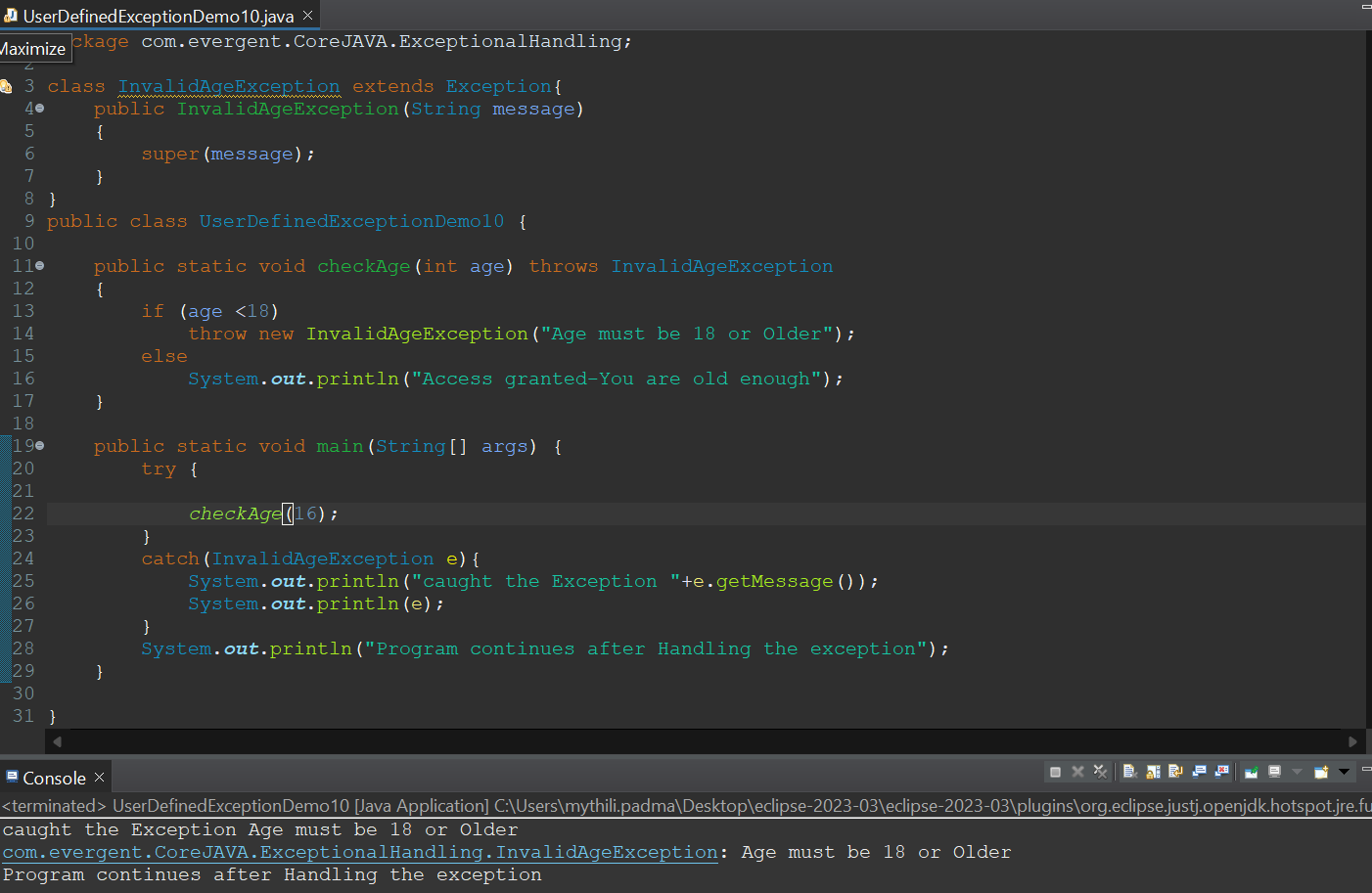


**Day-12 (21-08-2024)**

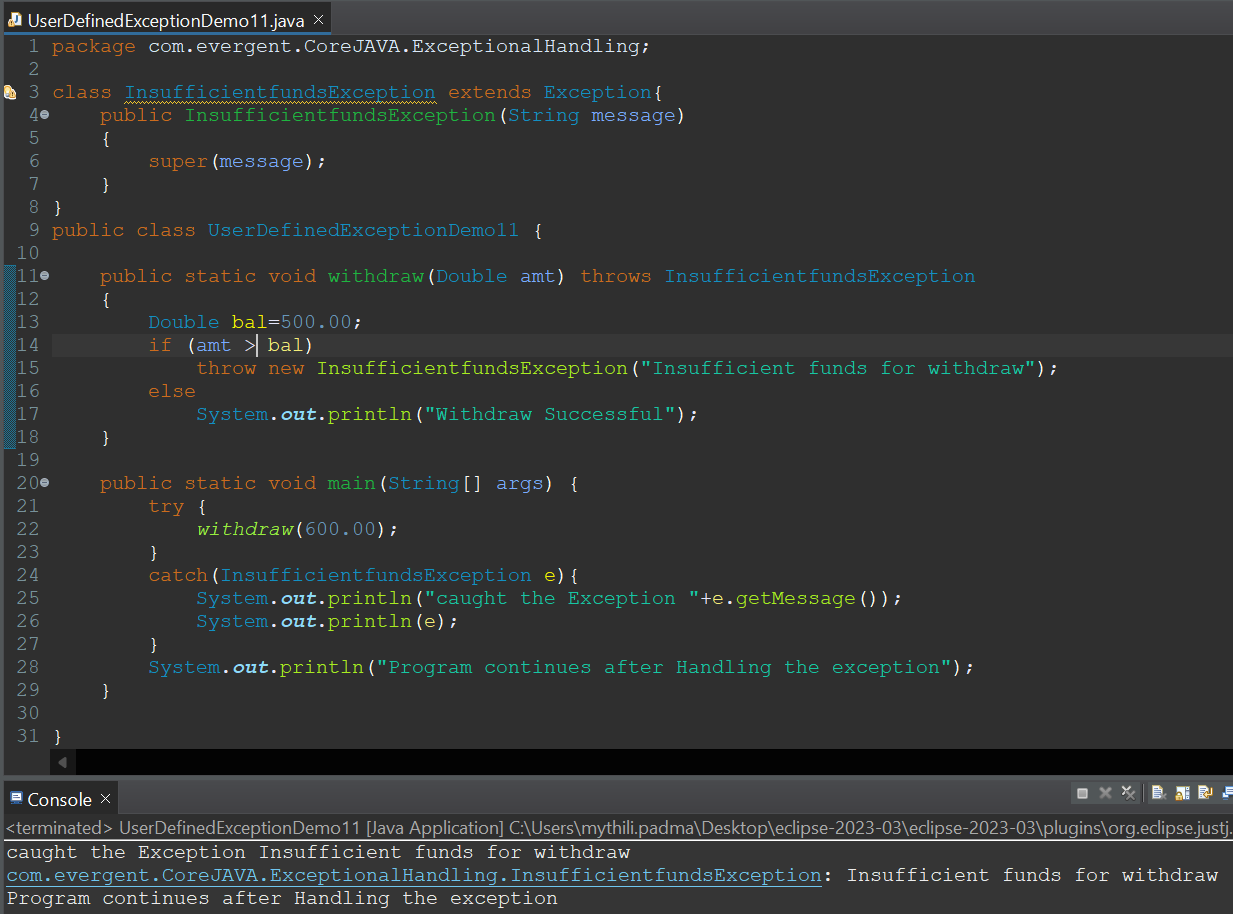
* Continuation of Exception Handling.
* User Defined Exception Handling.
* Unchecked Exception Handling.
* Runtime and Compile time Exception Handling.
* **Java Beans :**

1. Java Bean is a Mechanism.
2. Java Bean is light weight.
3. All attributes are private,get/set methods are public implements java.io.Serializable interface.
4. We can achieve tightly Encapsulation through java beans

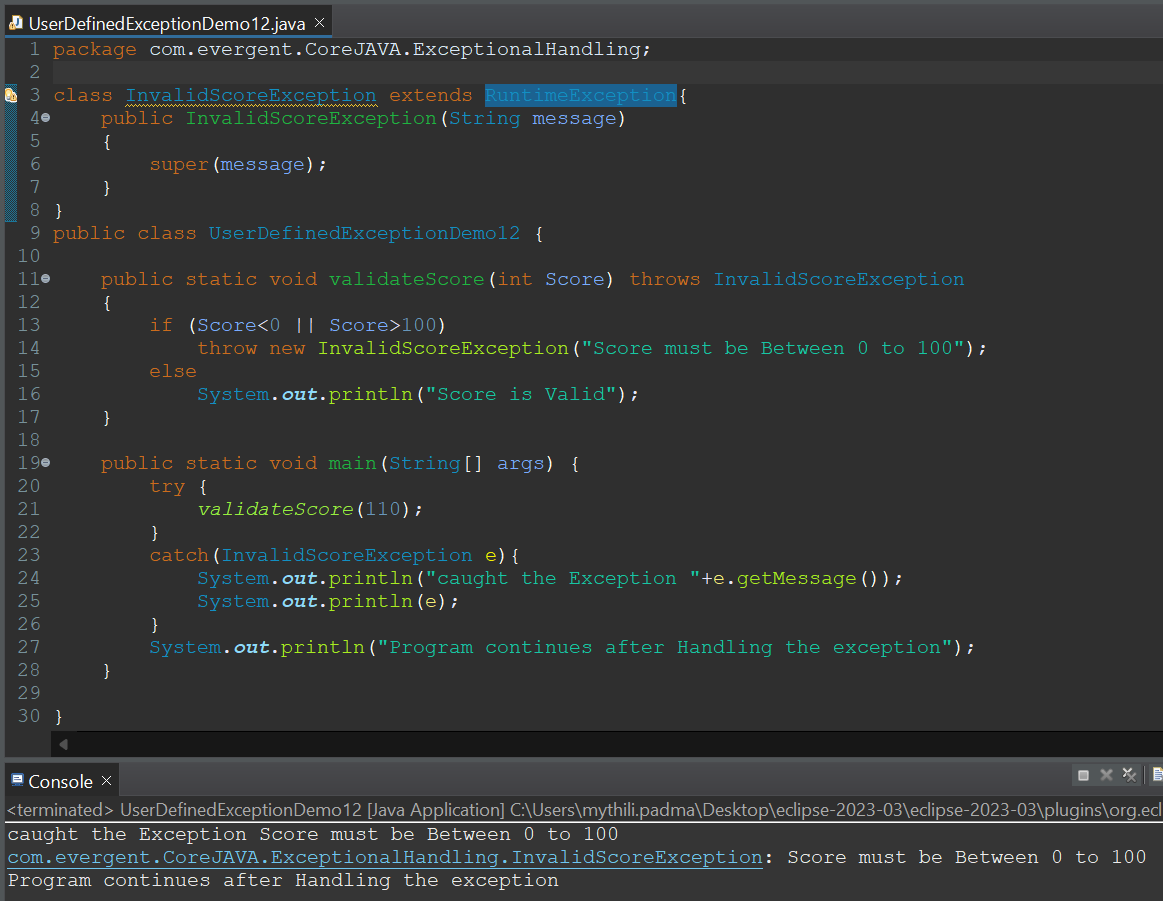
**Program 1:**

****

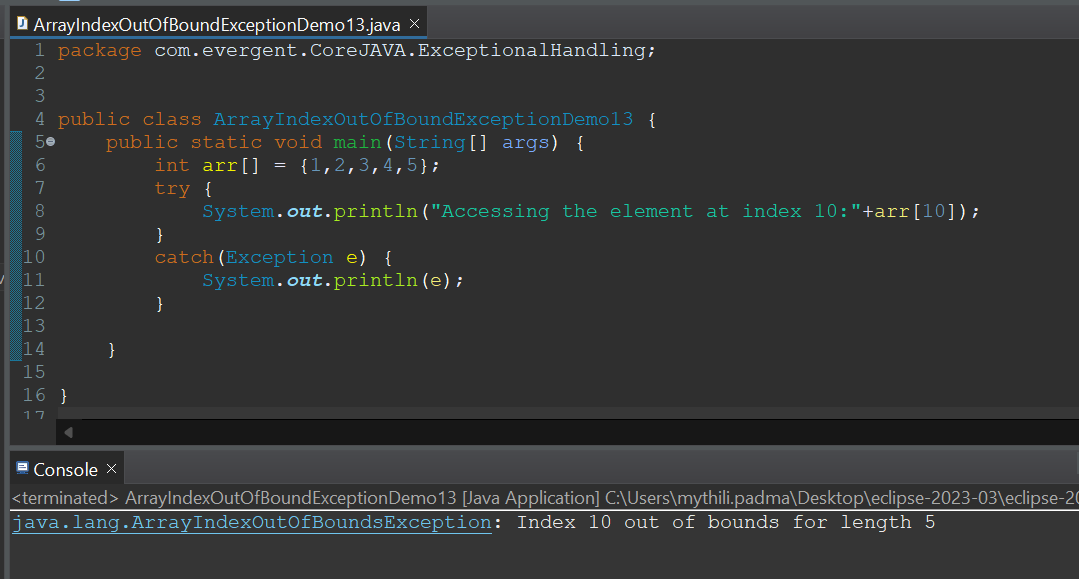
**Screenshot 2024-08-21 140317Program 2:**

****

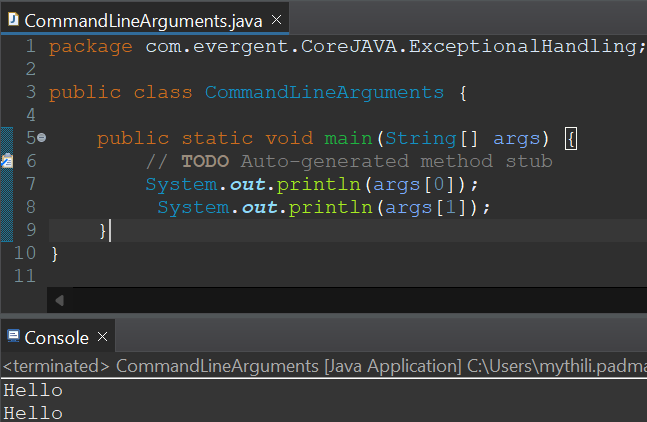
**Program3:**

****

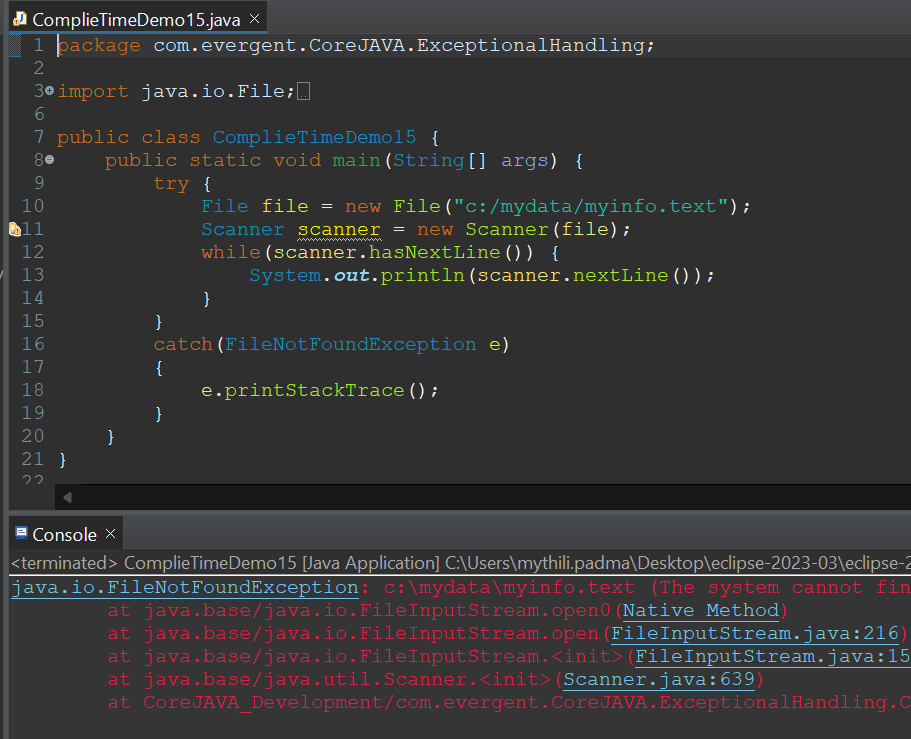
**Program 4:**

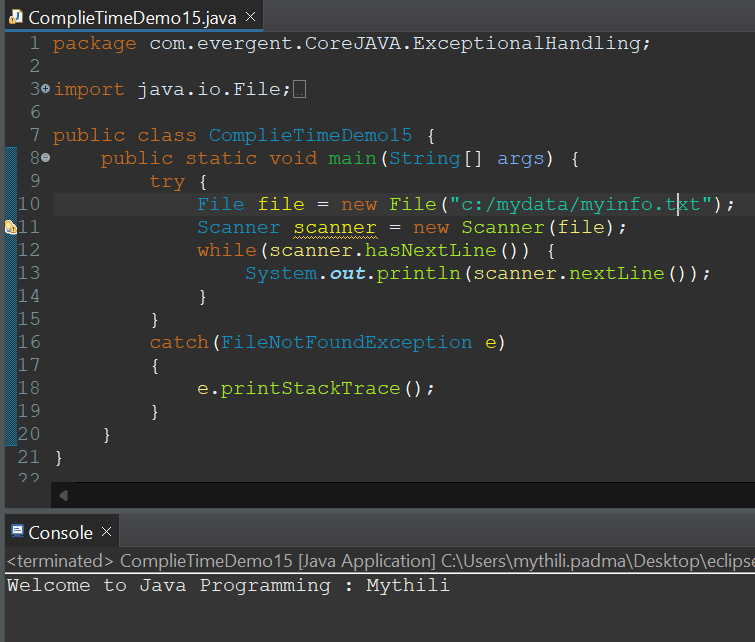
****

**Program 5:**

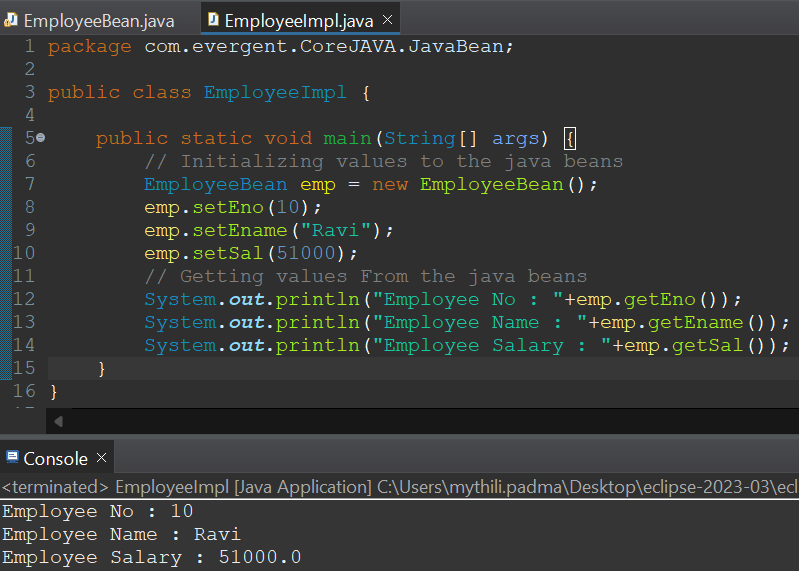
****

**Program 6:**

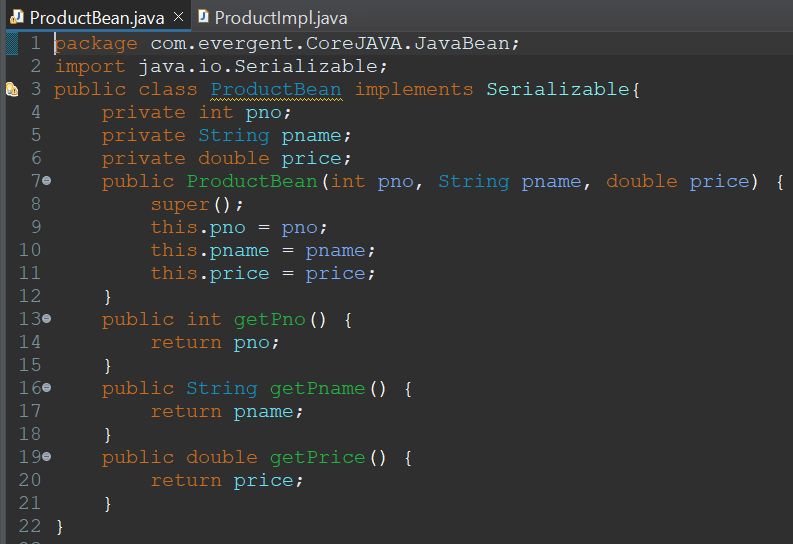
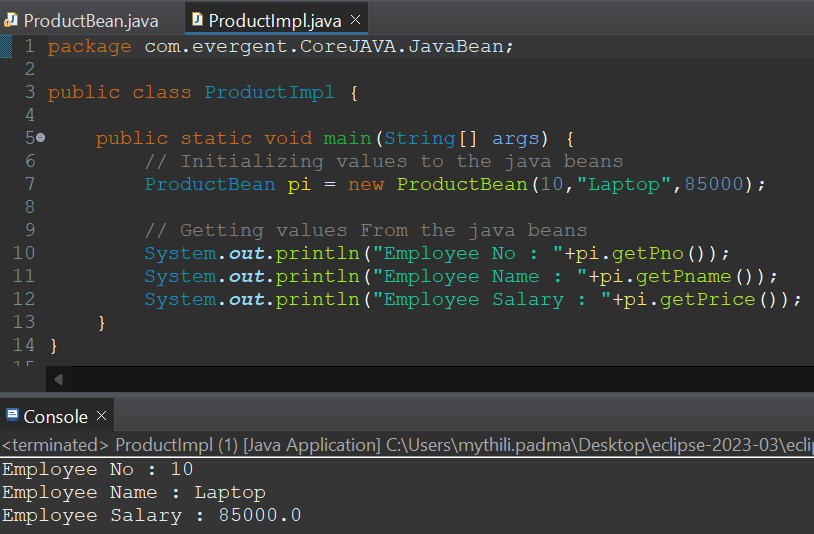
****



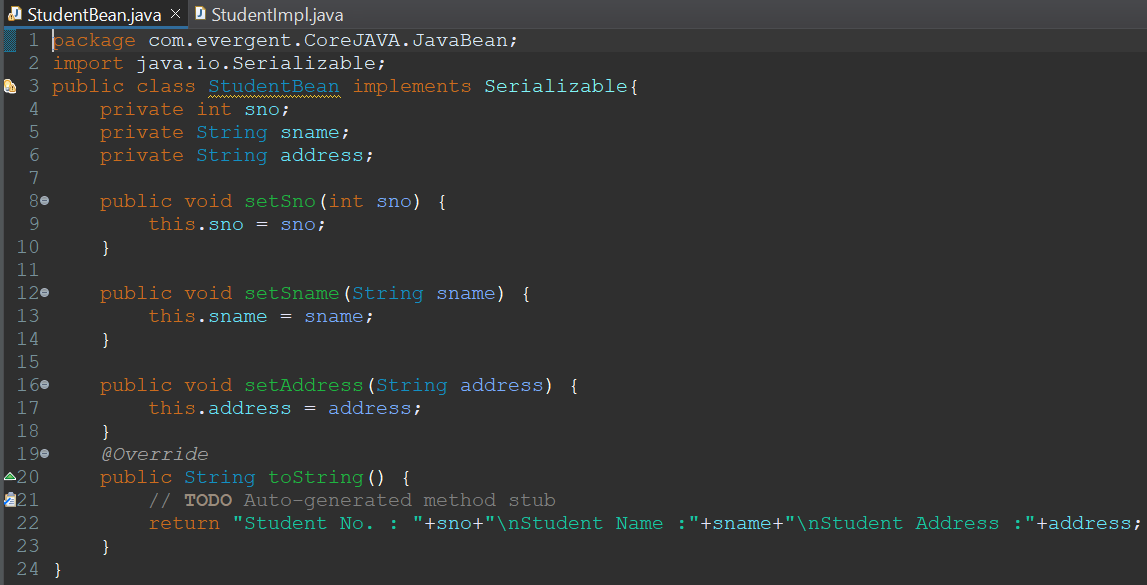
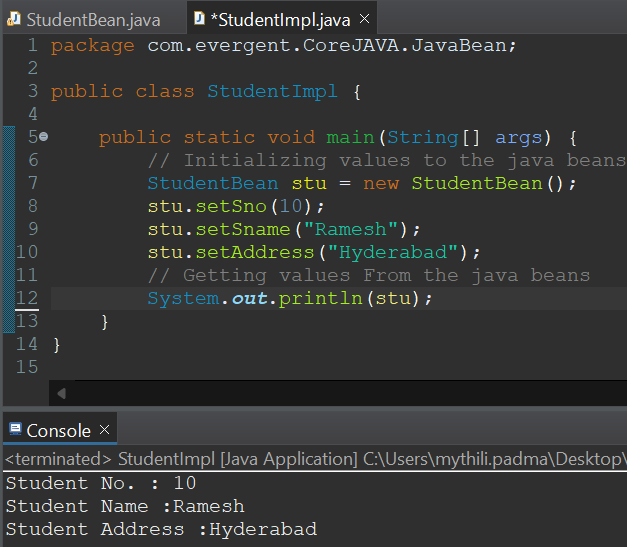
**Program 7: (JavaBeans )**

** **

**Program 8:**

** **

**Program 9:**

** **

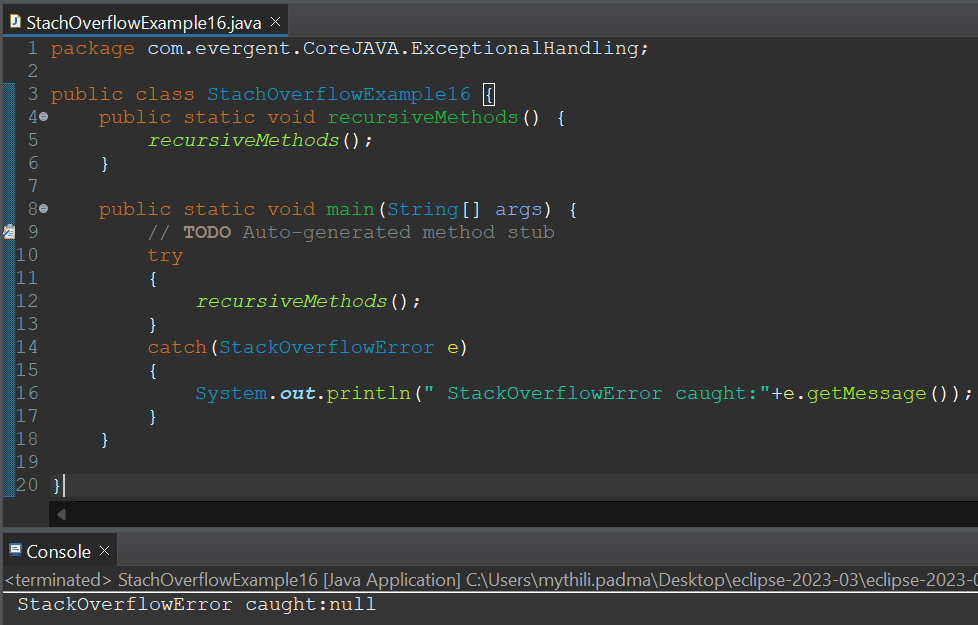
**Day-13 (22-08-2024)**

* **Revised Exception Handling**
* **Git &github**
* **Collections FrameWork:**

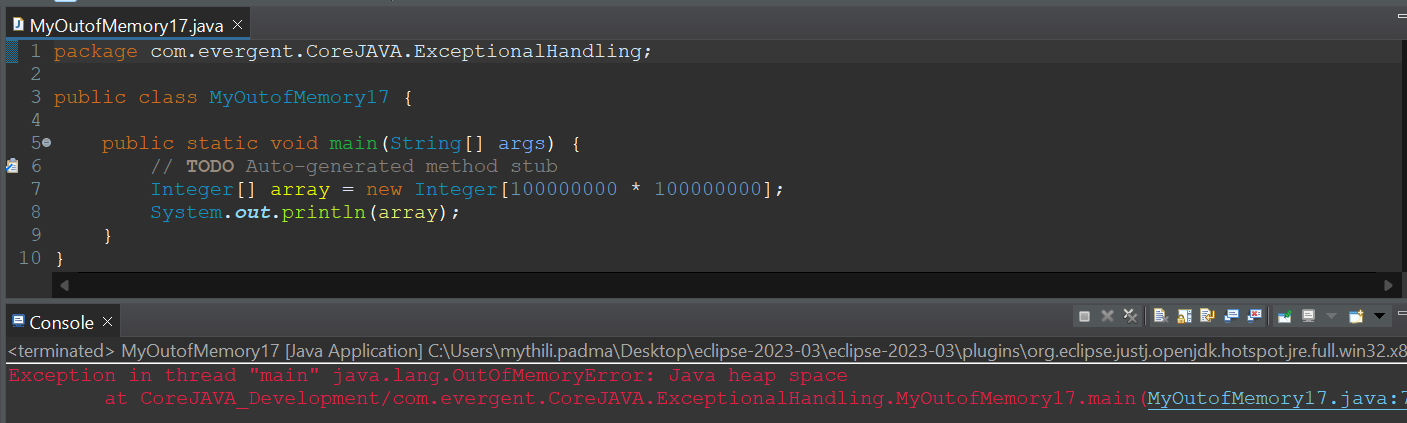
1. **<set>---> Set items are stored randomly and wont allow duplicates In Set, there are hasSet,TreeSet**
2. **<List>---->List itemsare stored Sequentially and will allow duplicates**

**In list, there are ArrayList,LinkedList**

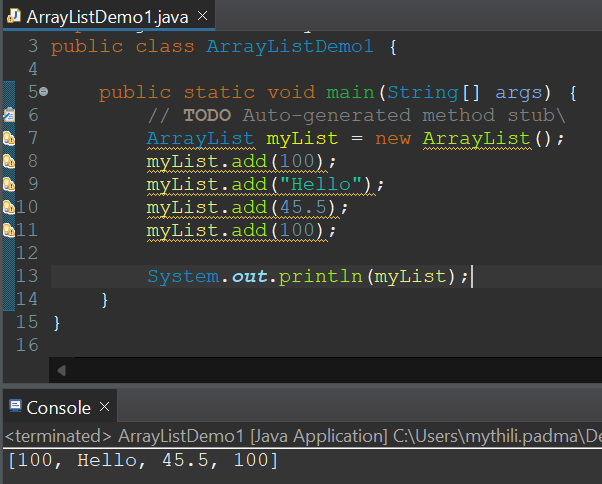
**Program 1:**



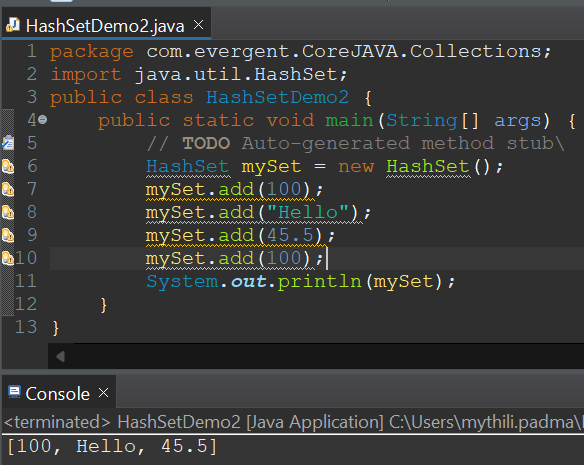
**Program 2:**



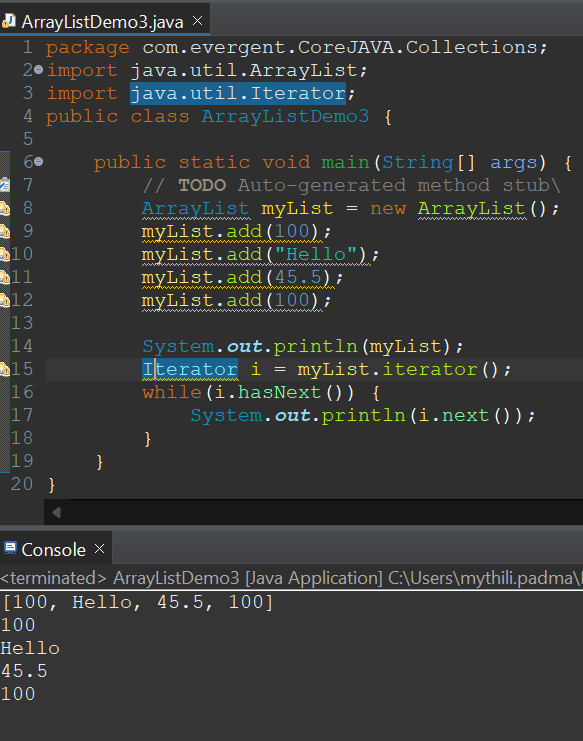
**Program 3: (Collections Framework)**

****

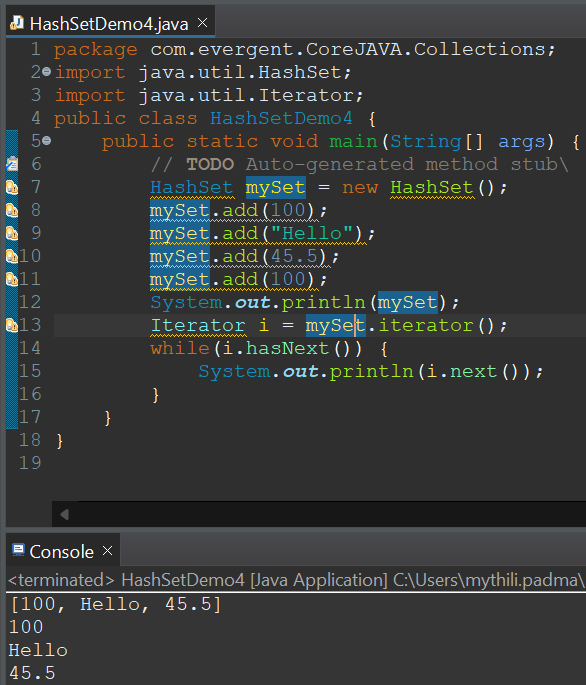
**Program 4:**

****

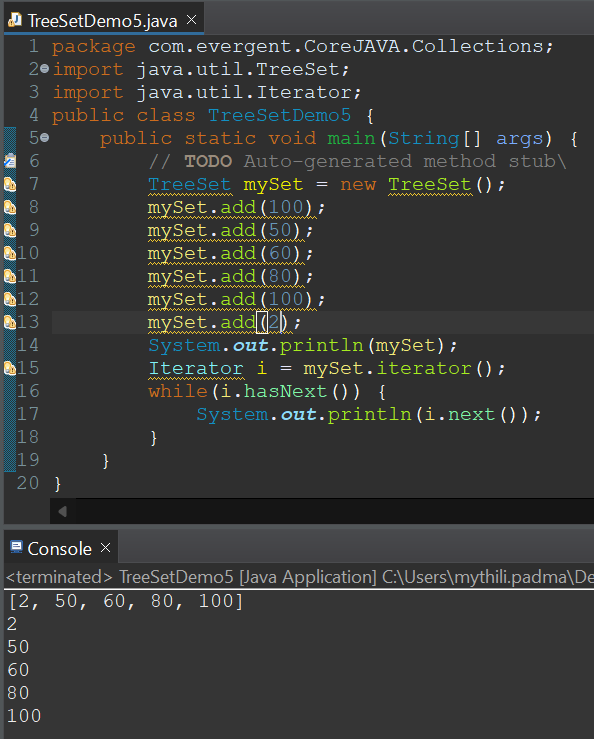
**Program 5:**

****

**Program 6:**

****

**Program 7:**

****