**Core JAVA training - index**

**Date:05/08/2024**

1. Language And Applications
2. JAVA Features

* Why Java is Independent?
* Oops
* Exception Handling
* Multi threading
* Web Application
* Open Source
* Security
* Support Networking
* Memory Management

1. JDK,JRE,JVM
2. Basic Java Programming
3. Packages

**Date:06/08/2024**

**Morning(11:00 am)**

1. Nested Loops
2. One Dimensional Array
3. Two Dimensional Array
4. Logical Programming

**After Noon(3:30 pm)**

1. SwitchCase
2. Scanner Class
3. Java.lang

* Object Class Methods

1. Enum
2. Event Management Application

**Date:07/08/2024**

**Mrng(11:00 am)**

1.oops

* Encapsulation

Programs

Calculation

Person

Method Flow

**After Noon(3:30 pm)**

1. Inheritance
2. Polymorphism

**.**Method overloading

**.**Method Overriding

1. Abstraction
2. IS-A (Inheritance)
3. HAs- A (Object Creation).

**Date:08/08/24**

**Constructor**

i. Class name and constructor name should be same

ii. There are 2 types of constructors

a. Default Constructor

b. Parameterized Constructor

iii. We can access constructor while creation of object

iv. Constructors are mainly for initializing

v. 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

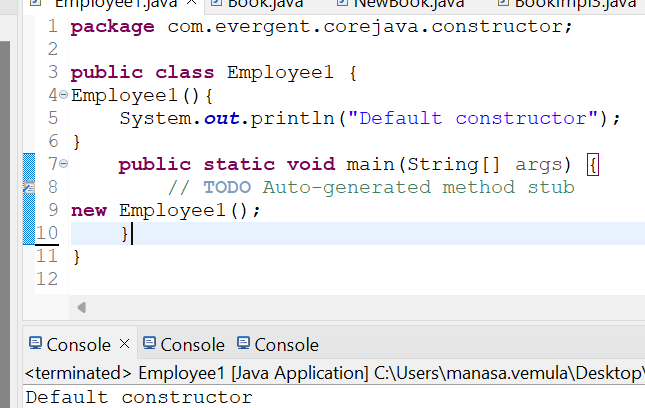
vi. Every class needs atleast 1 default constructor

vii. this, super This

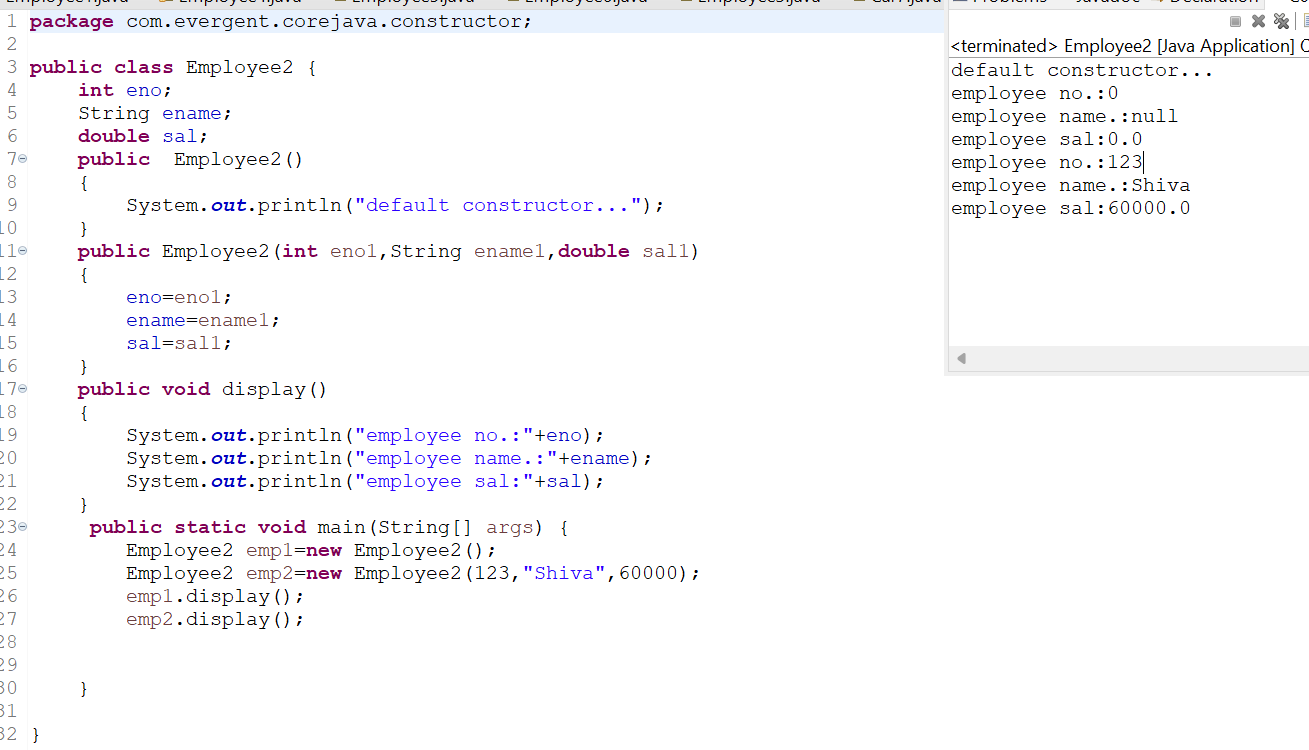
--> this is a keyword always refers to instance variables

viii. Always constructor are overloaded

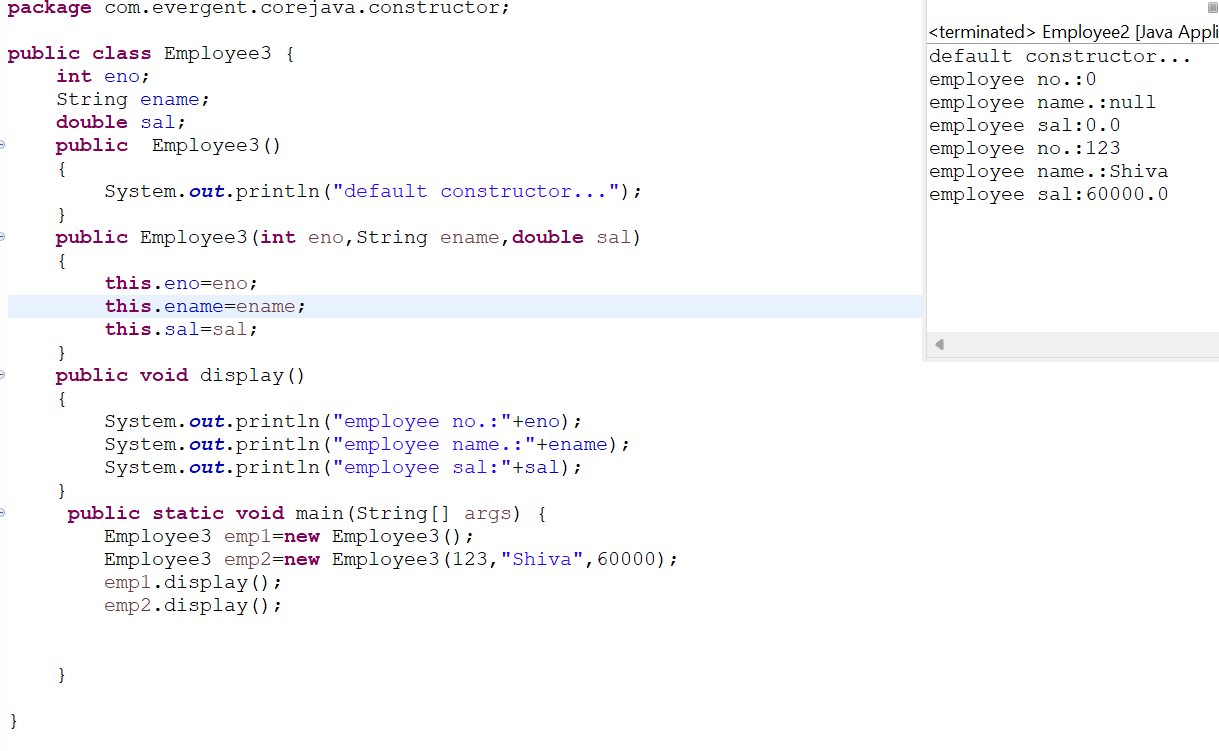
**Program1:**



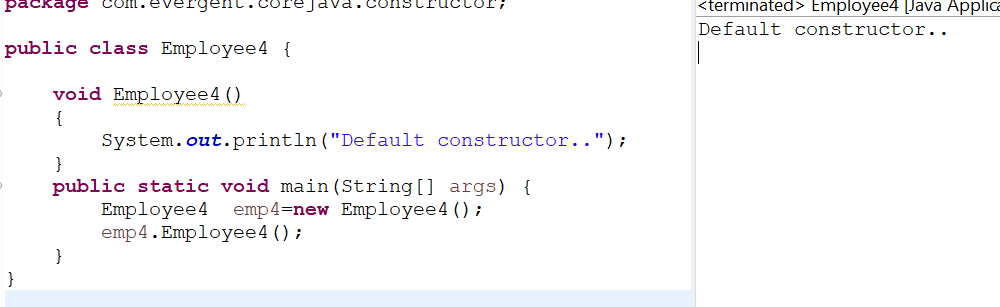
**Program2:**

****

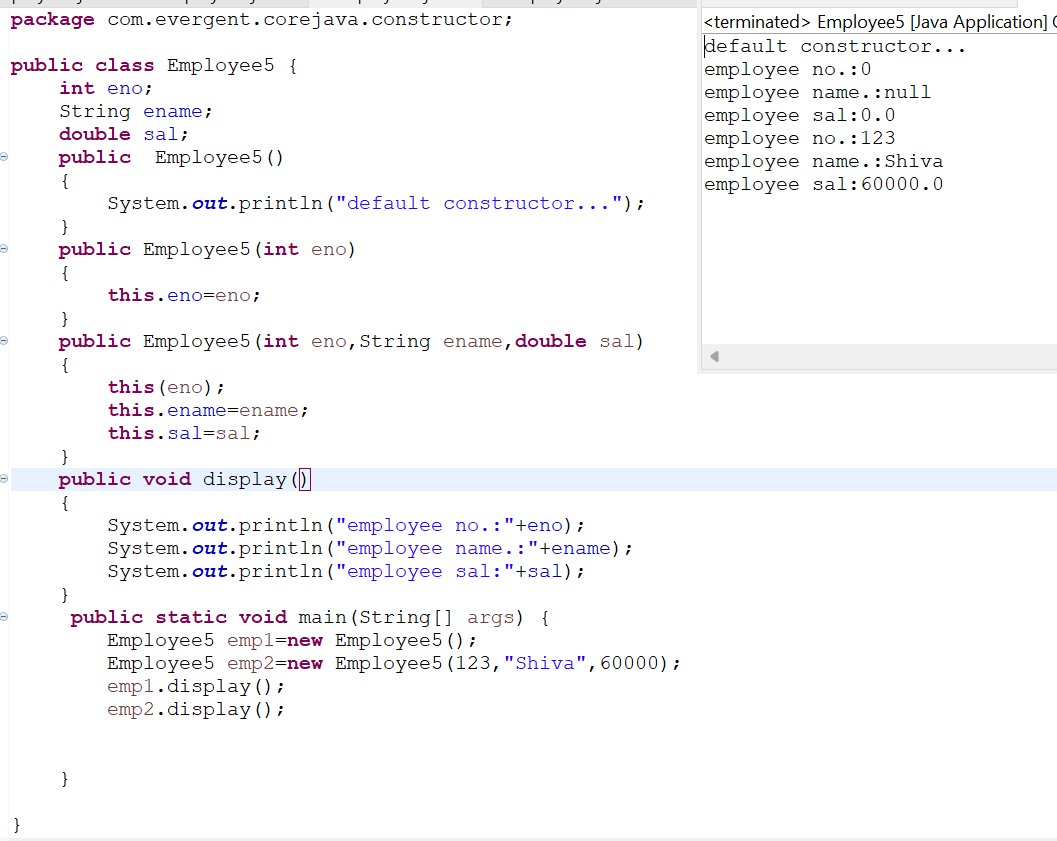
**Program3:**

****

**Program4:**

****

**Program5:**

****

**Program 6:**

****

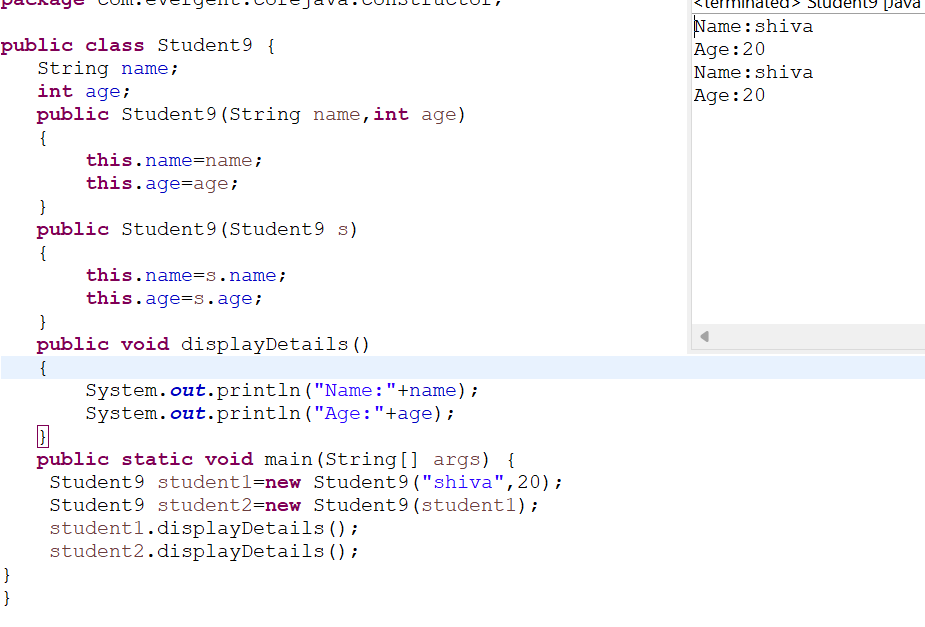
**Program 7:**

****

**Program8:**

****

**Program9:**

****

**Date:09/08/2024 - Day5**

1. **Static**

a. Static is a keyword

b. We can declare variables and methods as static

c. We can access static variables and static methods directly through calssname.methodname and classname.variablename respectively.

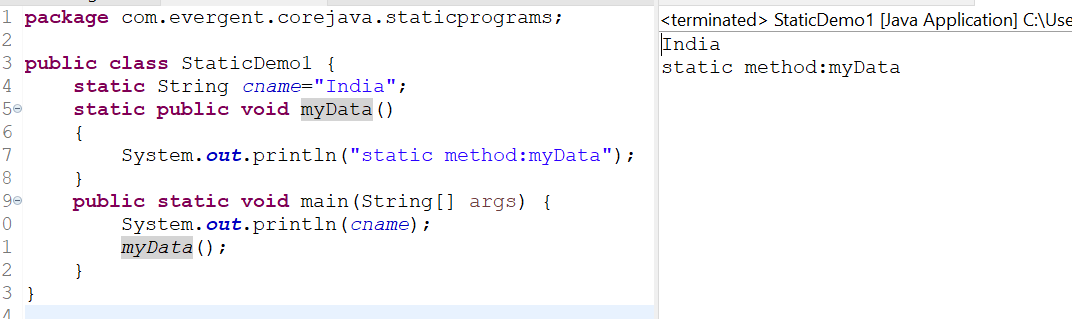
d. Static methods can access static methods and static variables only.

e. Static methods cannot access non static methods and non static variables.

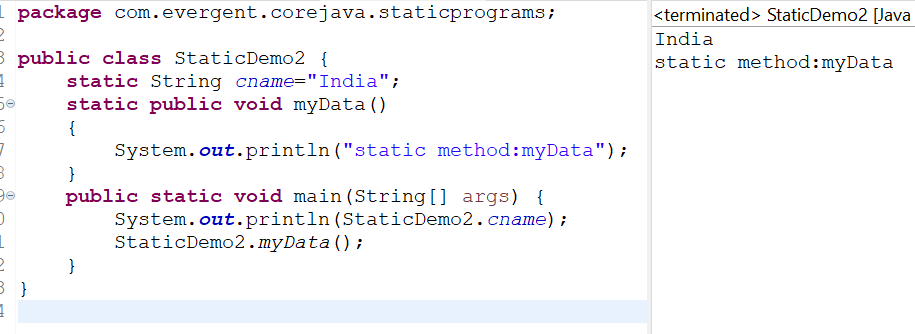
f. Non static methods can access static methods and static variables.

g. Static block- whenever class is loaded inside the JVM at that time static block is initiated.

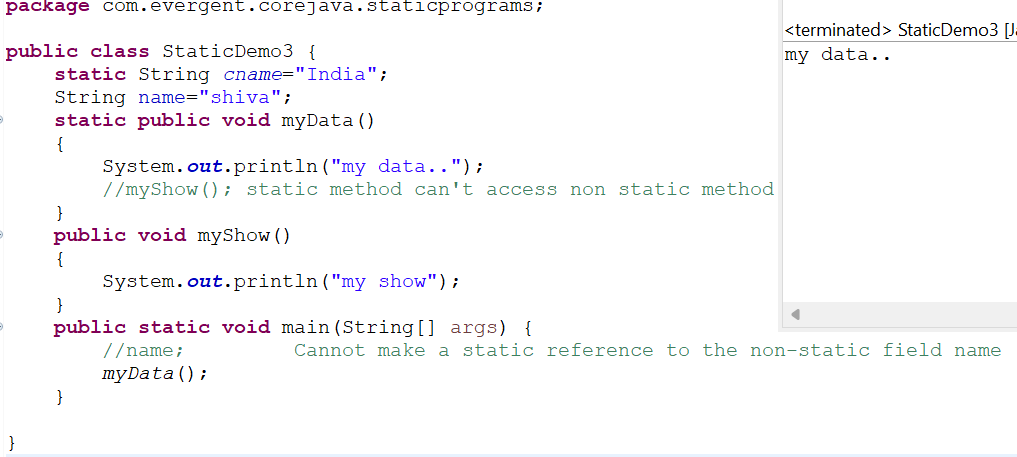
**Program1:**

****

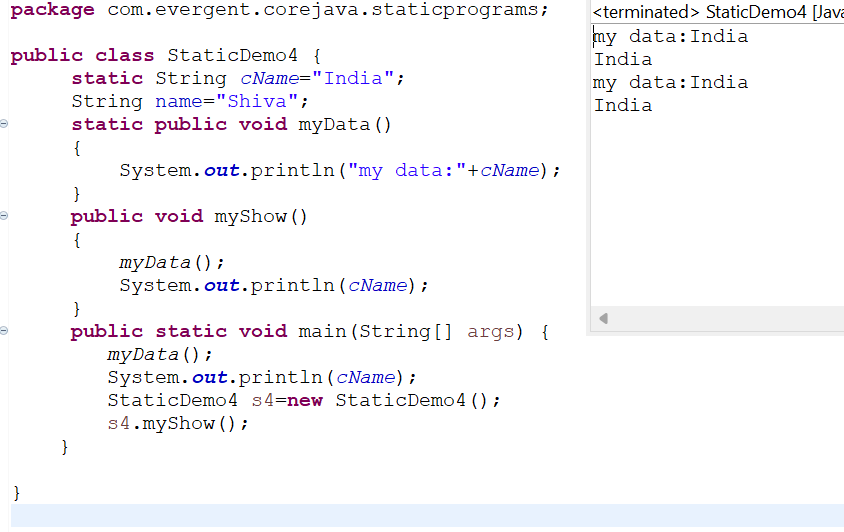
**Program2:**

****

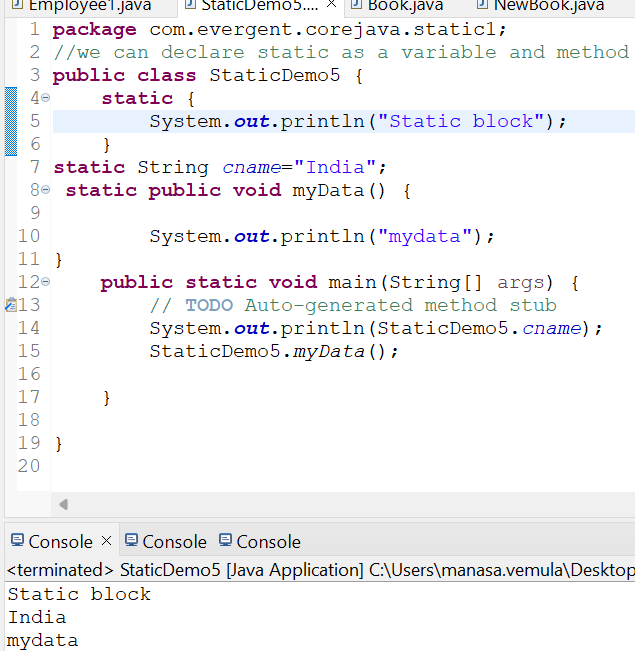
**Program3:**

****

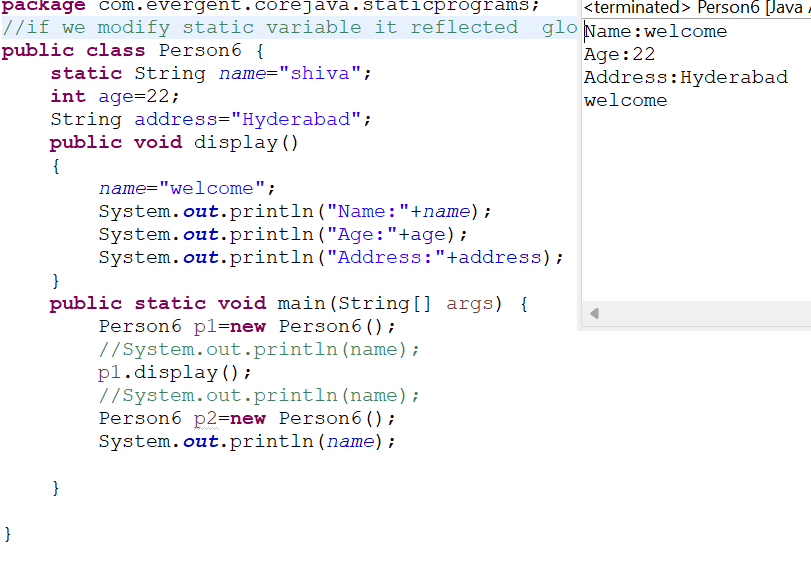
**Program4:**

****

**Program5:**



**Program6:**

****

**2. Final**

a. Final is a Keyword.

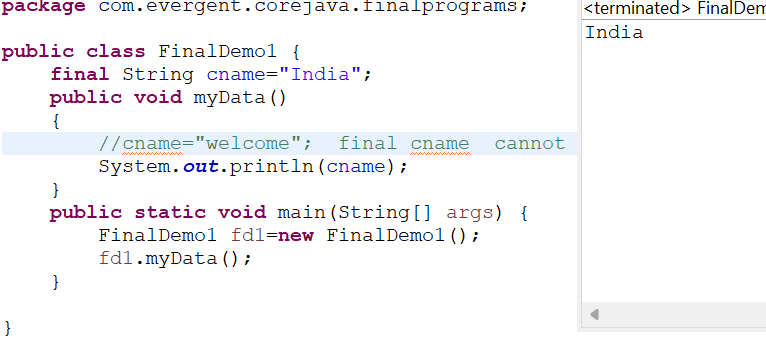
b. We can declare a variable, method, or a class as final.

c. Final variable cannot be modified.

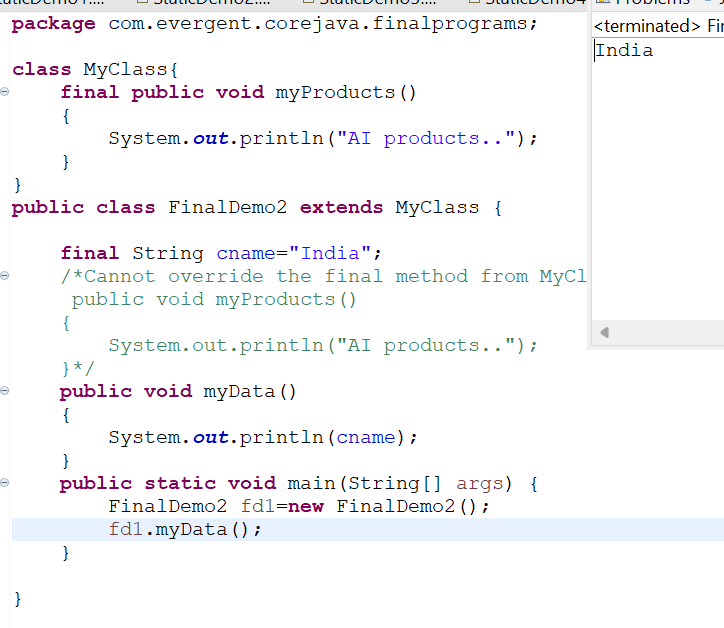
d. Final Method cannot be overrided.

e. Final class cannot be inherited.

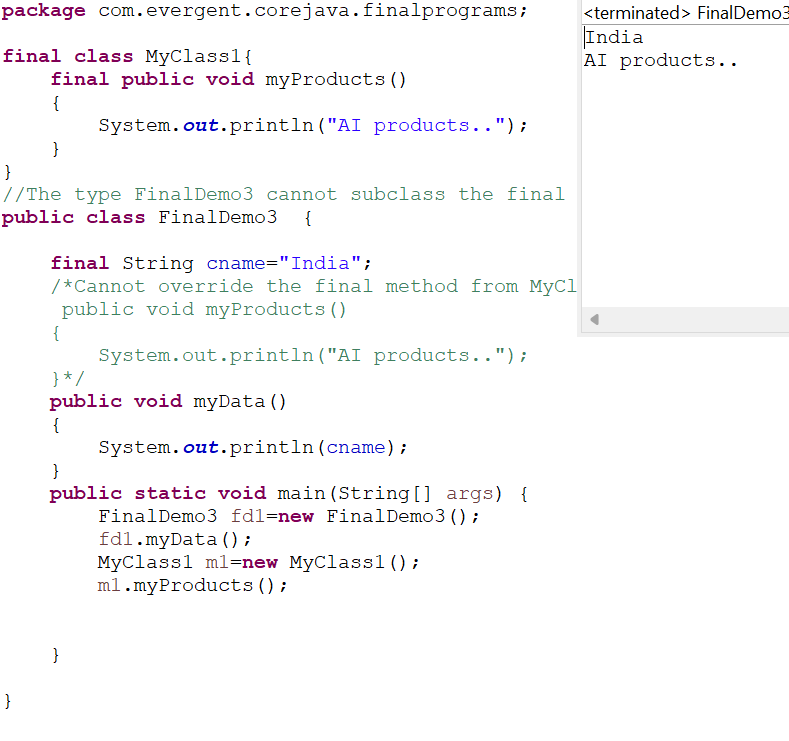
**Program1:**

****

**Program2:**

****

**Program3:**

****

**12/08/24 :**

**Strings:**

**-Why string is immutable?**

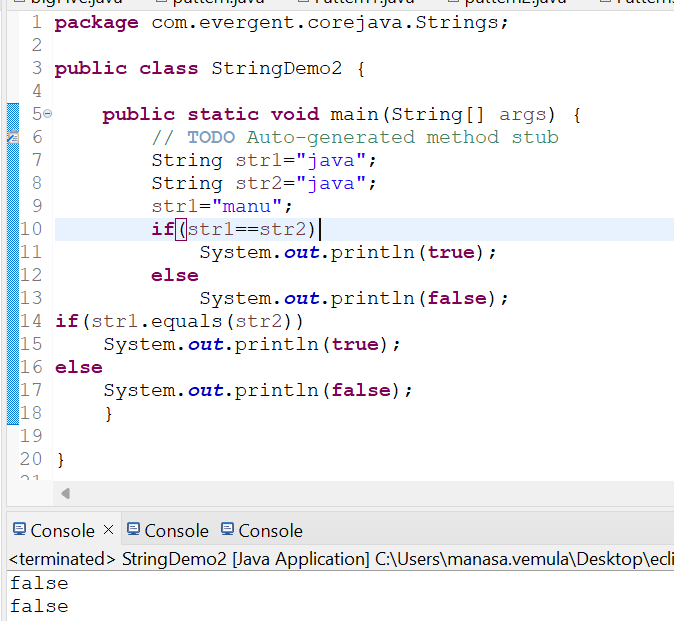
1. String is a final class
2. Strings are immutable
3. Strings having methods
4. All methods are non-synchronized

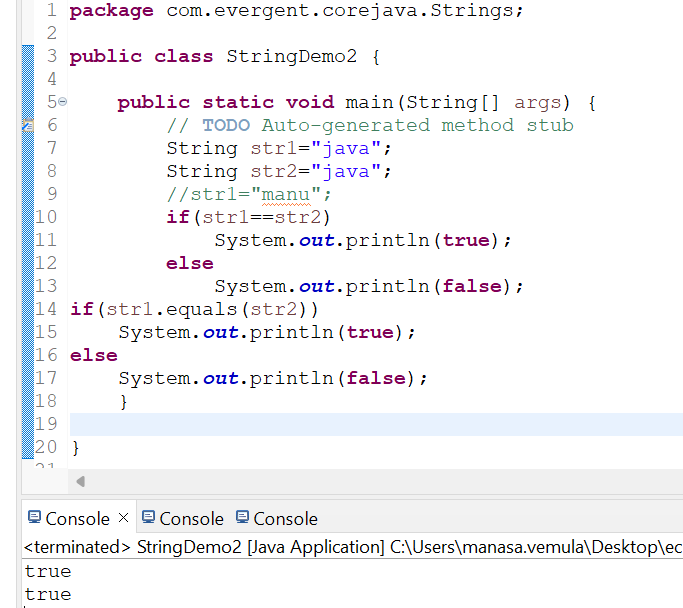
**StringBuffer:**

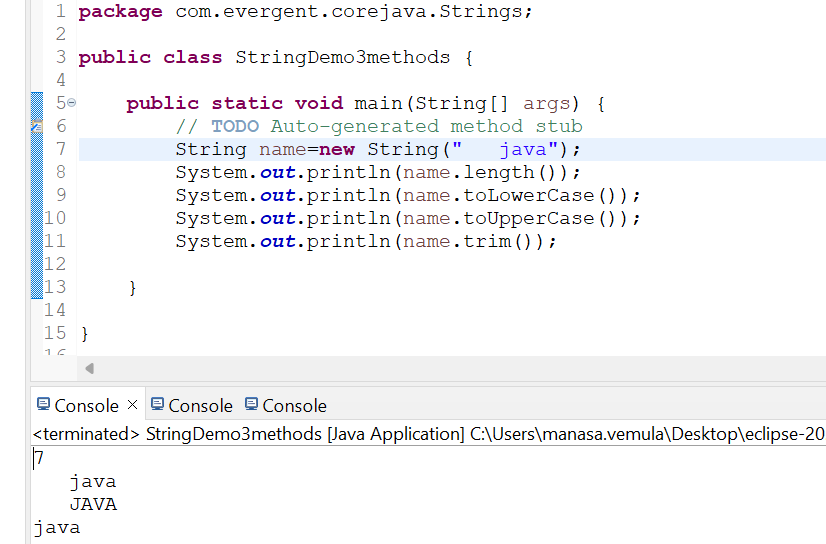
1. String buffer is a final class
2. String buffer is mutable
3. String buffer having methods
4. All methods are synchronized

**String Builder:**

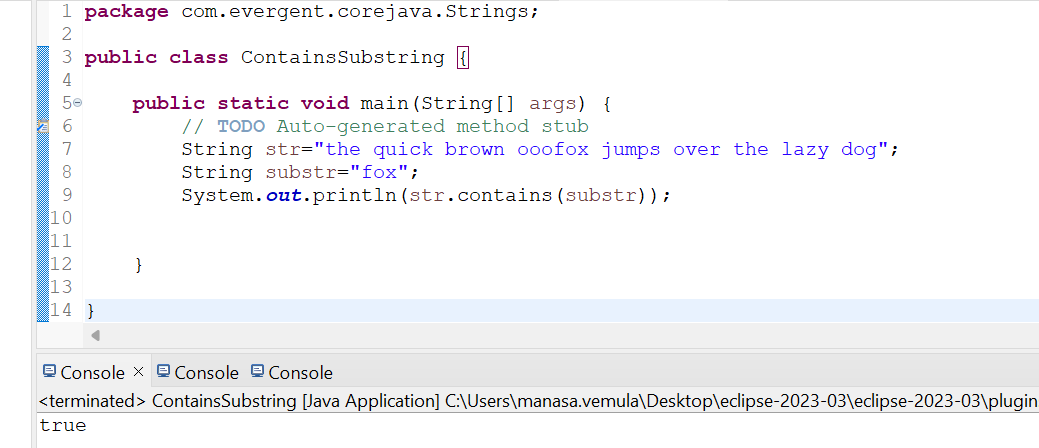
1. String builder is a final class
2. String builder is mutable
3. String builder having methods
4. All methods are non-synchronized



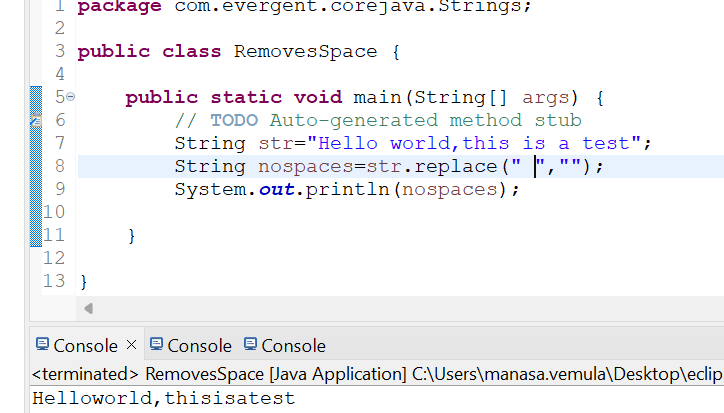




1.create a java program that creates a string and checks if it contains specific subString and then prints out the result

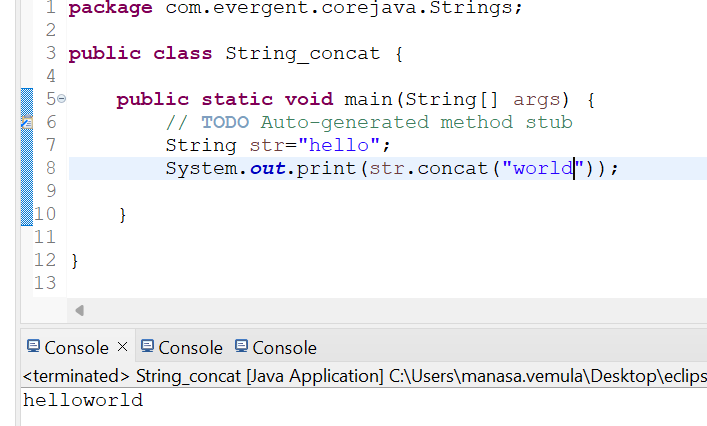


1. Write a java prgm to create a String ,remove all spaces from the string and then print out the result

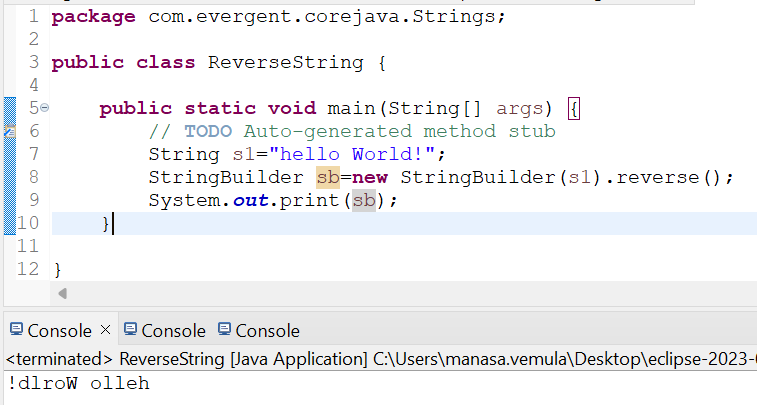


String concatination:

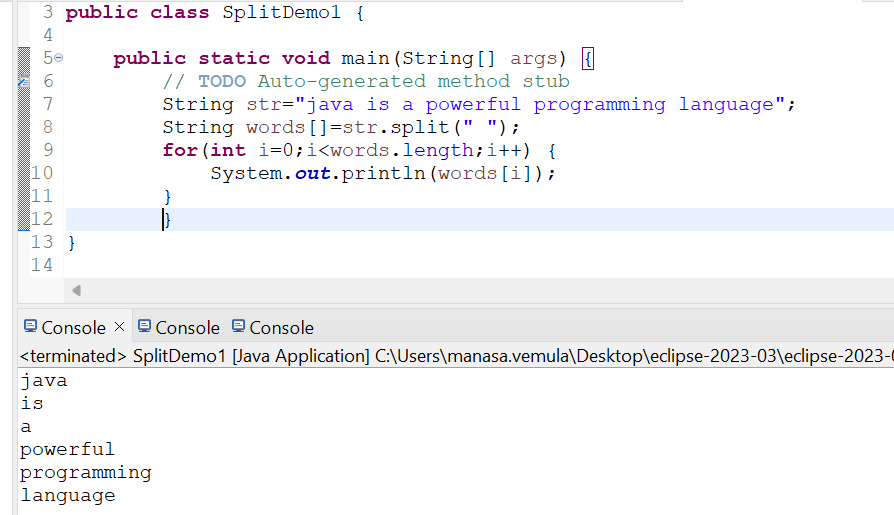
Strings can be concatenated using + operator (or) concat.



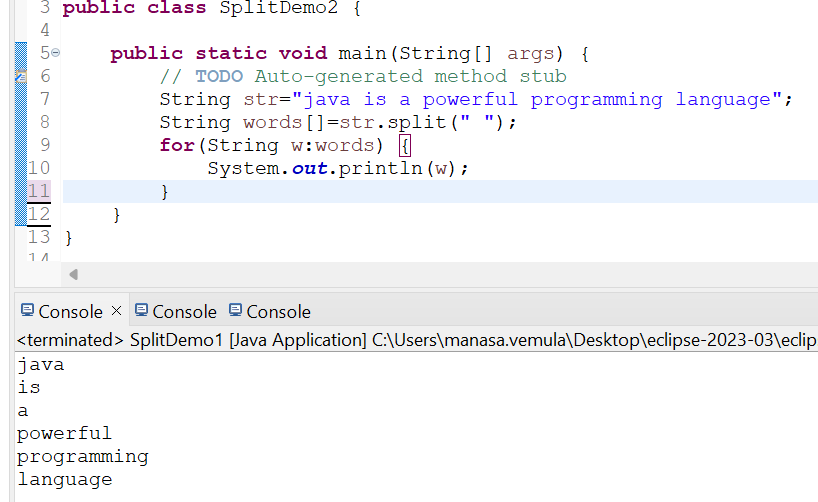
Reverse of a String:

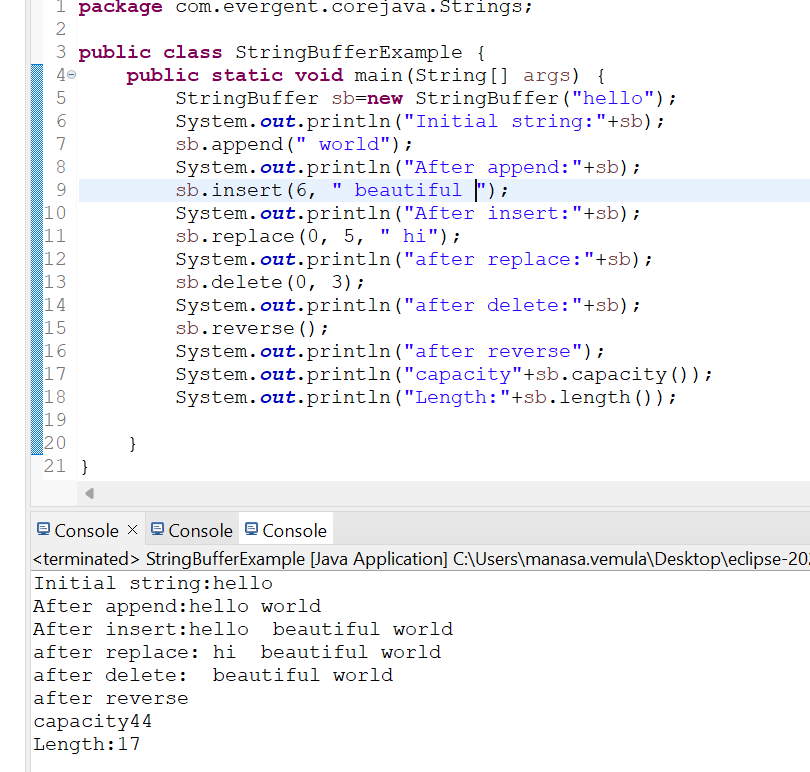


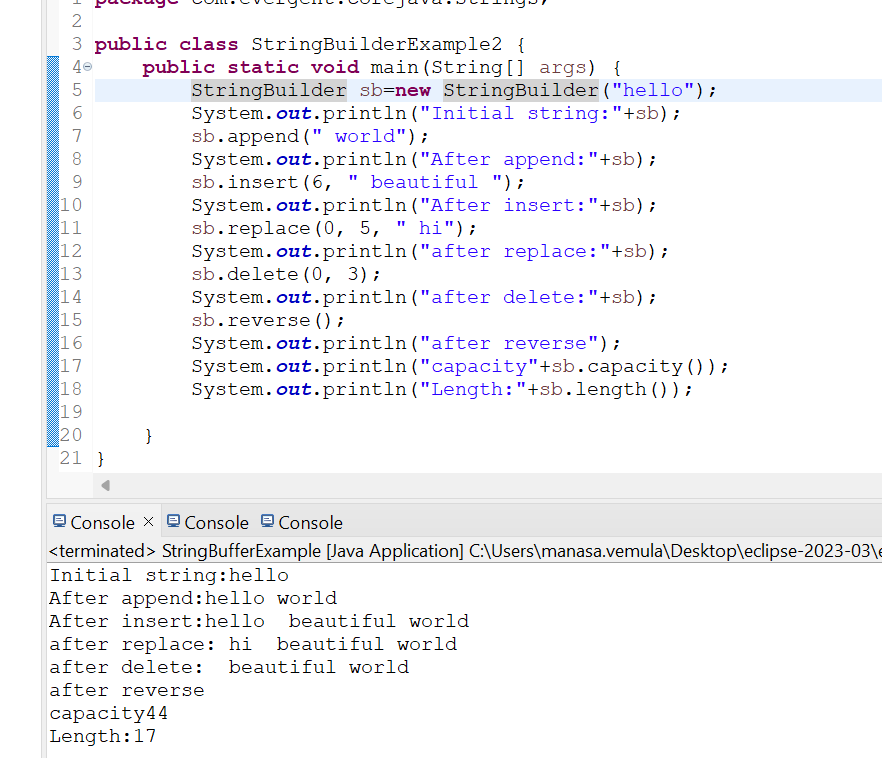
Splitting of any sentence (or) text on spaces



Using for each loop:



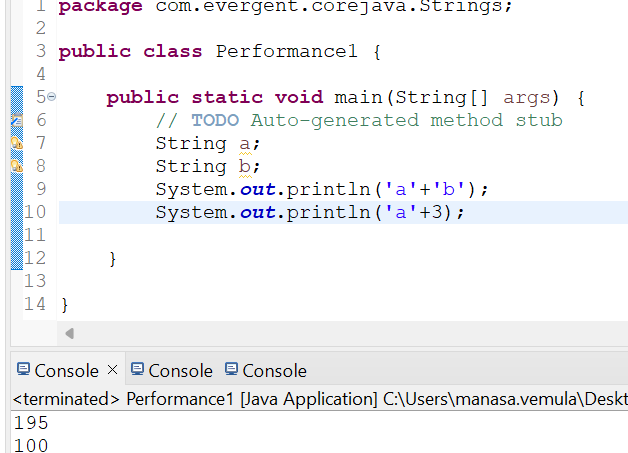


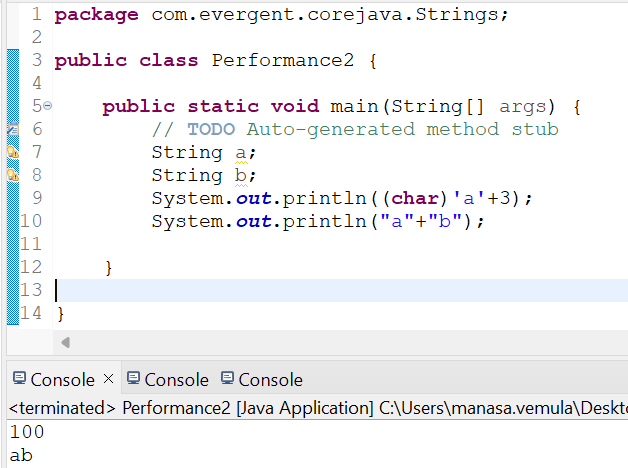


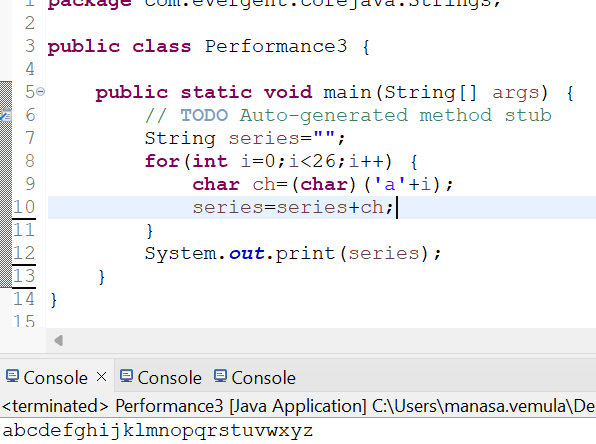
**String class important points:**

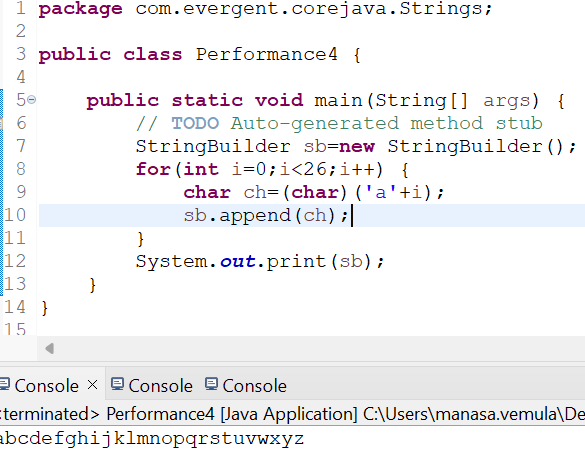
1. In java a string is a sequence of characters ,often used to represent text.
2. Strings are objects in java and are instances of the string class,which is part of the java java.lang package
3. Key features of strings in java:
4. Immutable:once a string object is created ,it cannot be changed .
5. Java optimizes memory usage by storing strings in special area of memory as string pool
6. If two strings have the same value and are created without using new keyword they will refer to same object in the stringpool.
7. We can create a string in java in multiple ways:
8. Using string literals :str=”hello world”;
9. Using the new keyword

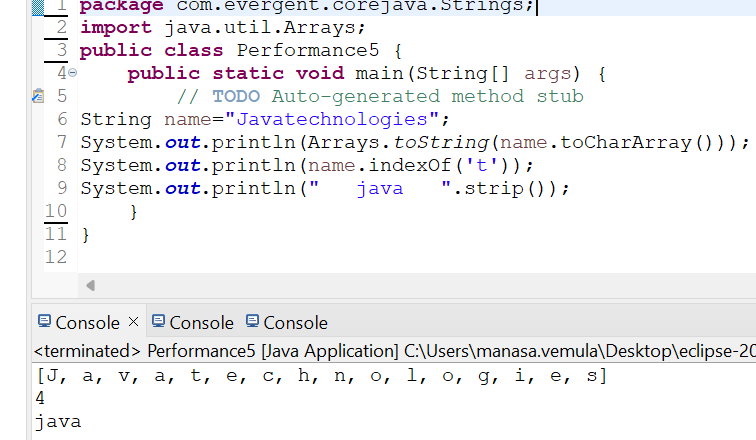
String str=new String(“hello, world”);











**13/08/24-Day7**

1. **Can you make class as immutable?**

We can make our class as immutable by declaring class as final and attributes are private and final

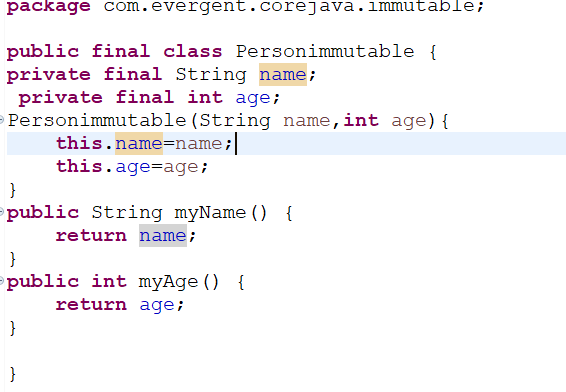
We can declare class as our own immutable class

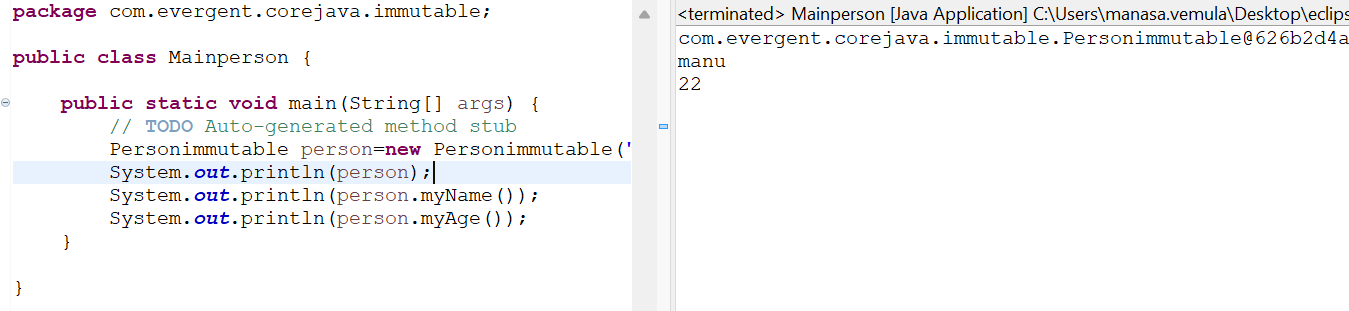
1. we can declare class as final
2. The classs is declared as final so that it cannot be subclass
3. Private Final class:
4. The fields name and age are private and final

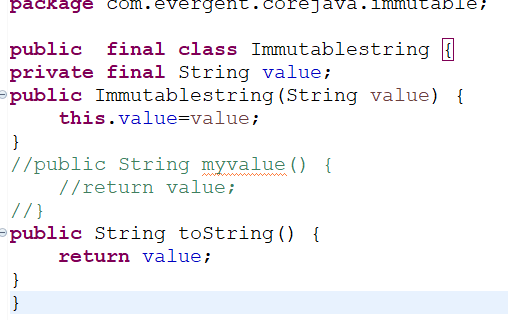
Constructor:

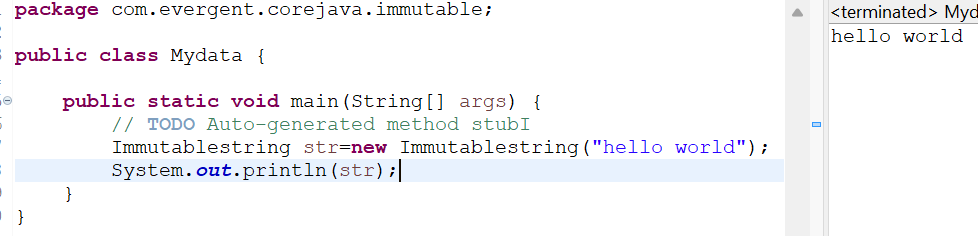
1. The constructor initialies the final fields when a person object is ceated

**Immutable class:**

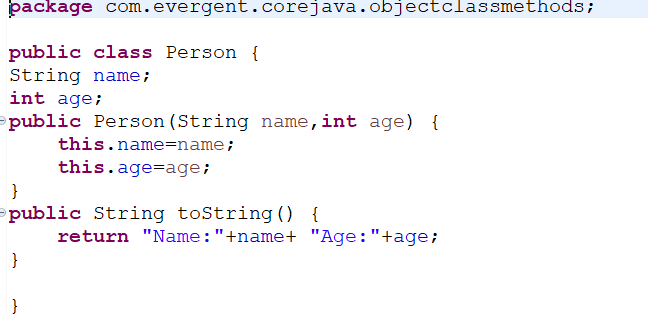


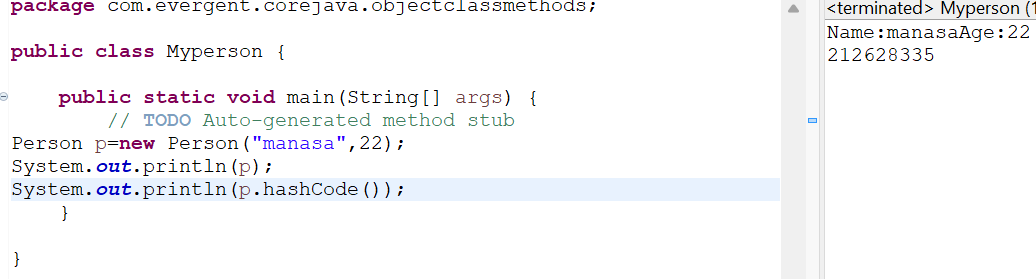






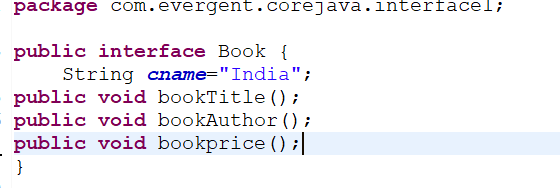
**toString and objectclassmethods:**

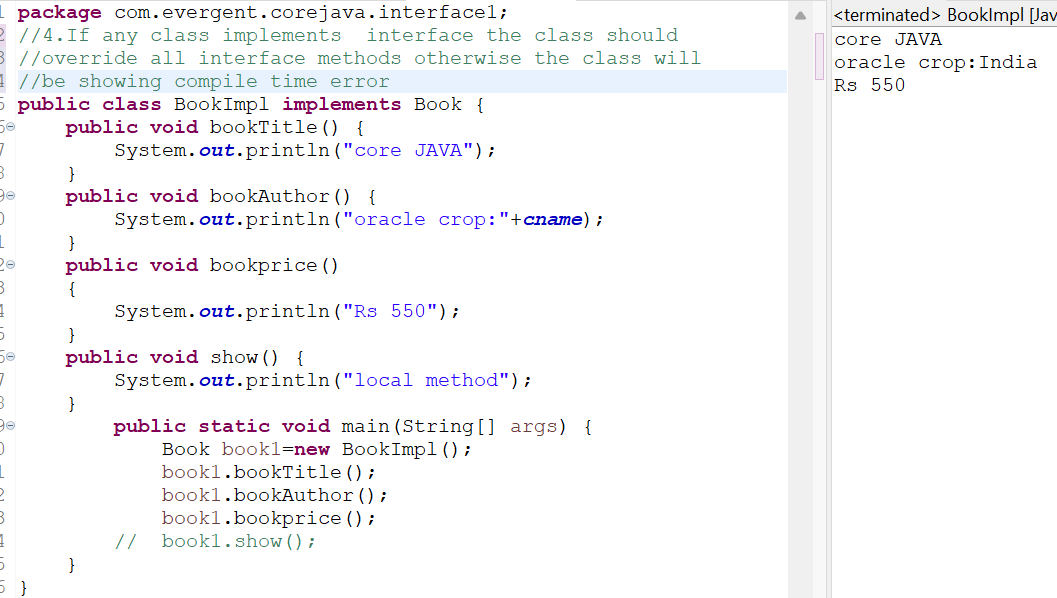


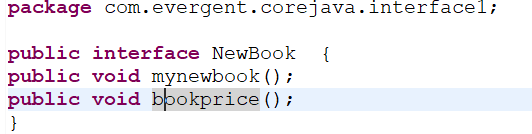


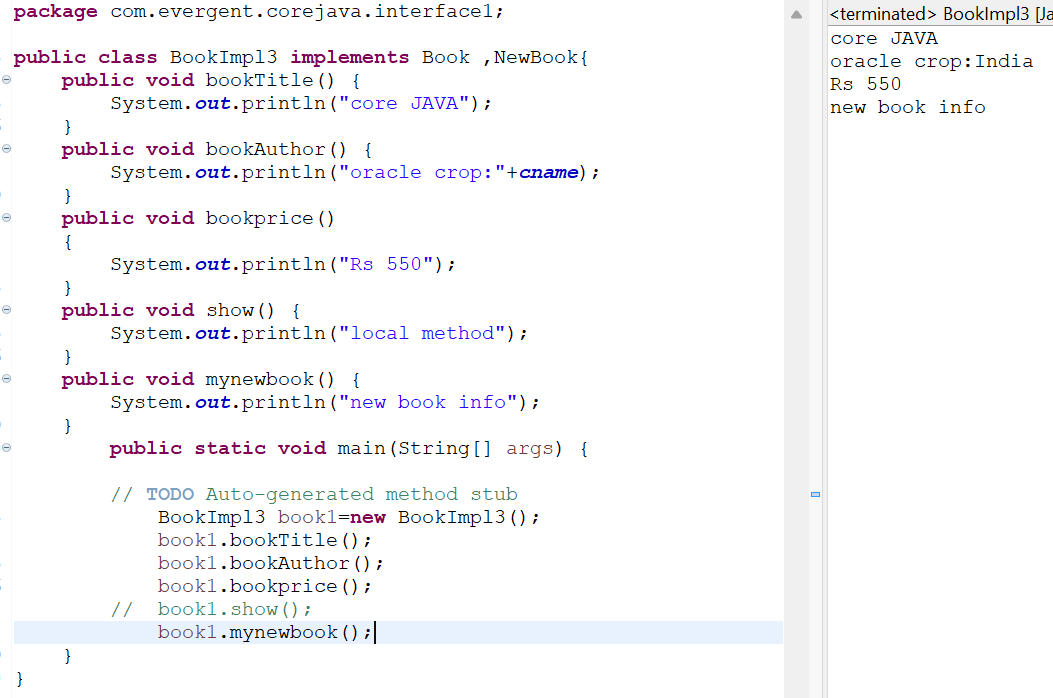
**Interfaces:**

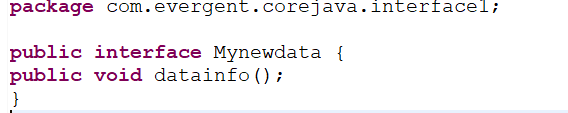
1. Interface is a keyword
2. We can declare method signature only but not implementation
3. By default all interfaces methods are abstract
4. We cannot create object but we can create reference to interface
5. By default variables inside interface is public static final
6. Java will support multiple inheritance through interfaces
7. One class can implements multiple interfaces
8. One interface can extend other interface

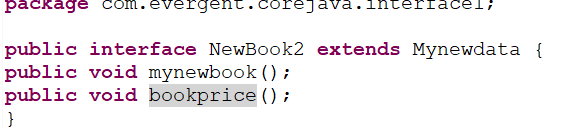


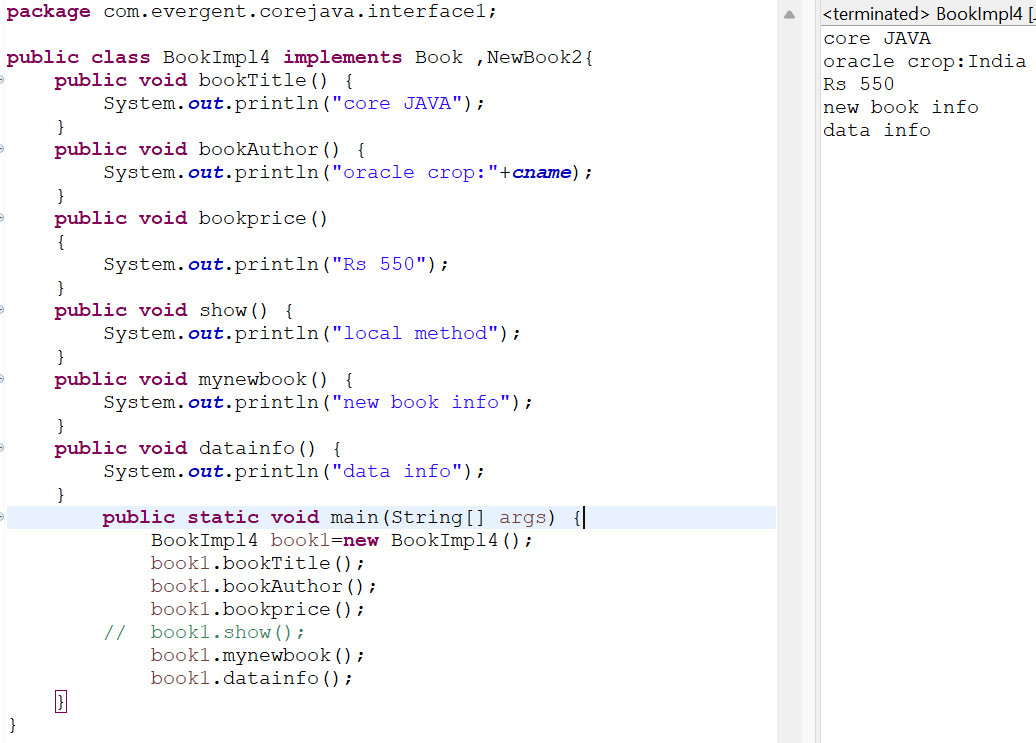












**Day-8 : 14/08/24**

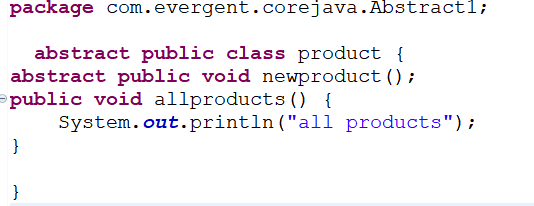
**Abstract Class:**

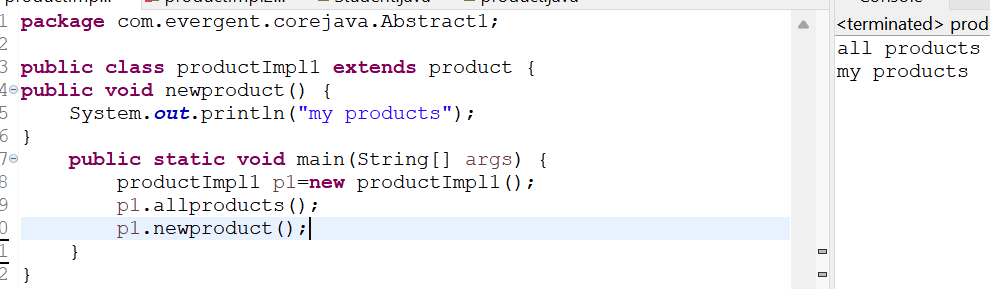
1. Abstract is a keyword
2. Abstract class having abstract methods and concrete methods (implemented)
3. If any class method having one abstract method the class should be declared as abstract keyword otherwise will be showing compile time error
4. If any class extends abstract class the class should override all abstract 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

-because it is not a fully implemented class

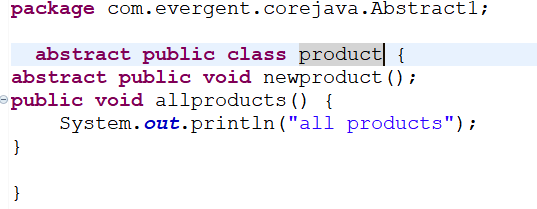
1. We can create constructor to abstract class
2. We can access abstract class constructor throw object creation

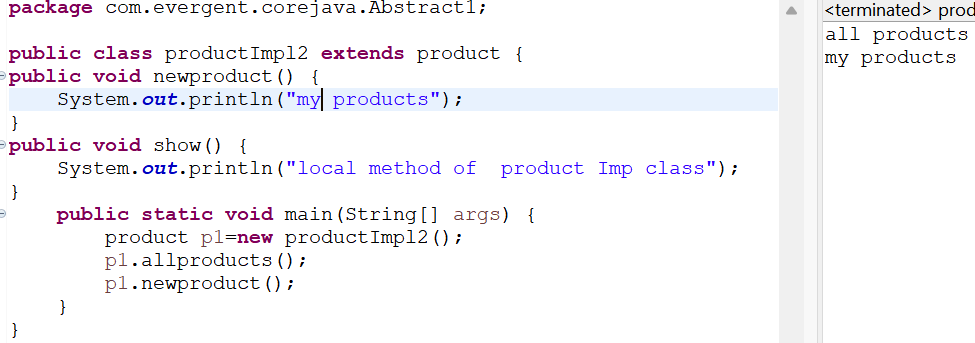
**Program 1:**



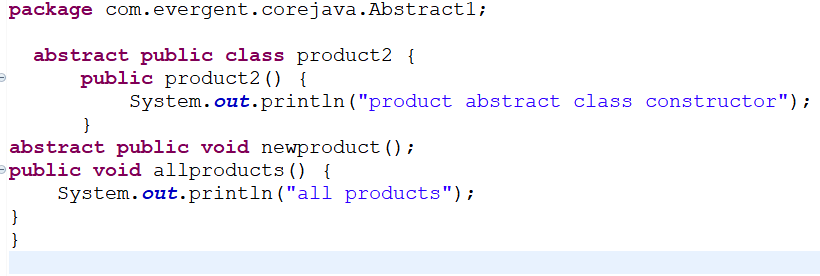


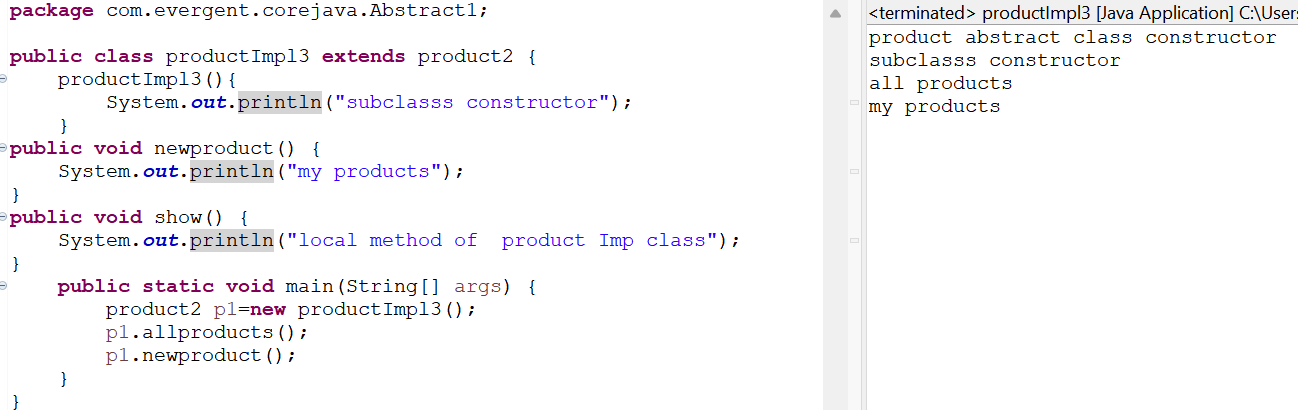
**Program 2:**





**Program 3:**

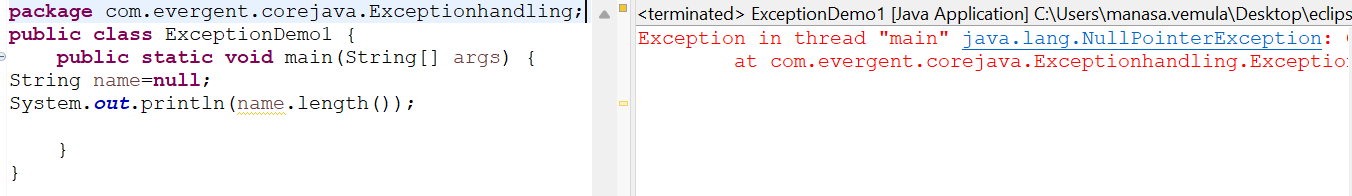




**Exception Handling:**

1. Exception handling is mechanism
2. Exception are inbuilt mechanisms of java
3. All exceptions are executed while abnormal conditions only
4. Normal flow it won’t execute any exceptions
5. Once any exceptions occurring in java then remaining lines of code is unreachable
6. Java.lang.Throwable is superclass for exceptions and errors
7. There are two types of exceptions in java:
8. Checked
9. Unchecked
10. All checked exceptions are compile time exceptions
11. All unchecked exceptions are runtime exceptions
12. There are five keywords in exception handling
13. Try
14. Catch()
15. Finally
16. Throws
17. Throw
18. Try is for business logic
19. Catch is for handling exceptions
20. Finally is a block,if exceptions is occur or not finally block will be executed
21. Throws an exception will be executed method by method
22. Throw is for runtime exceptions and will call predefined exceptions or userdefined exceptions
23. Try followed by either catch block or finally block
24. We should follow exceptions hierarchical
25. We create our own exceptions (user Defined)
26. Our own exceptions extends exceptions or runtime exceptions
27. All exception cases are into java.lang package
28. If there are two exceptions in class,Developer should handle one after another
29. Developer can’t handle errors

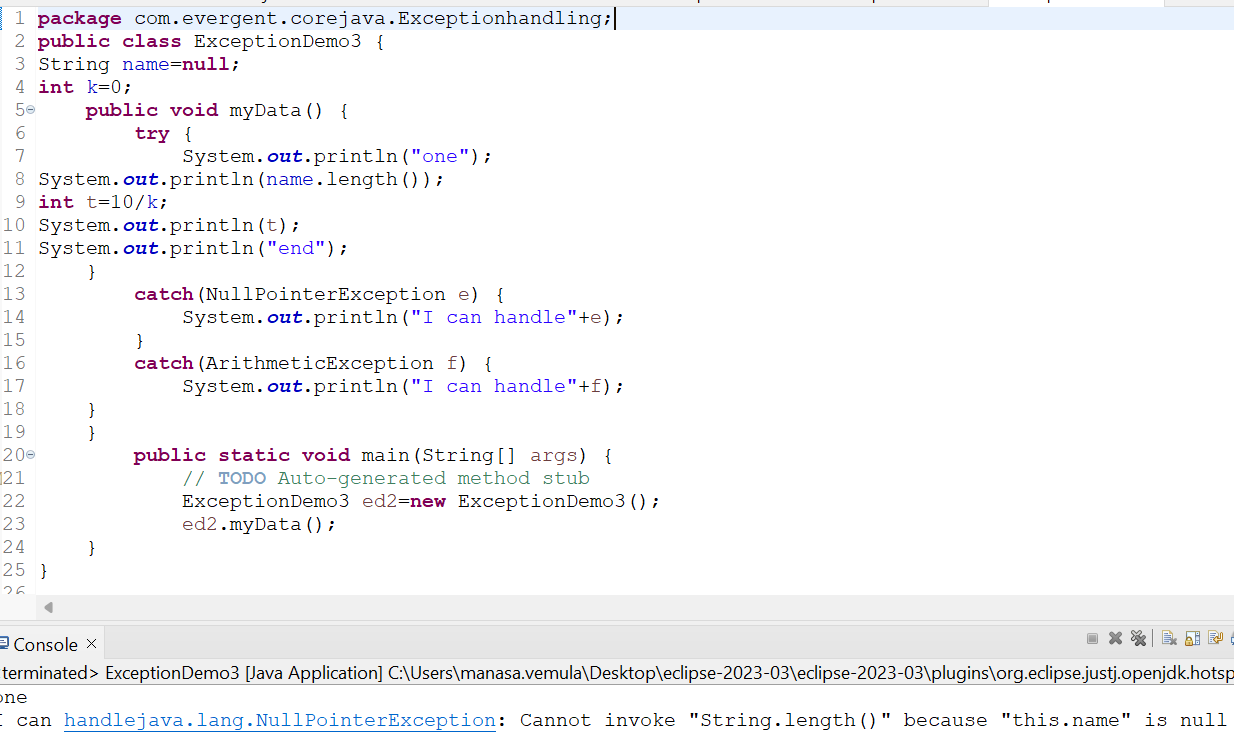
**Program1**



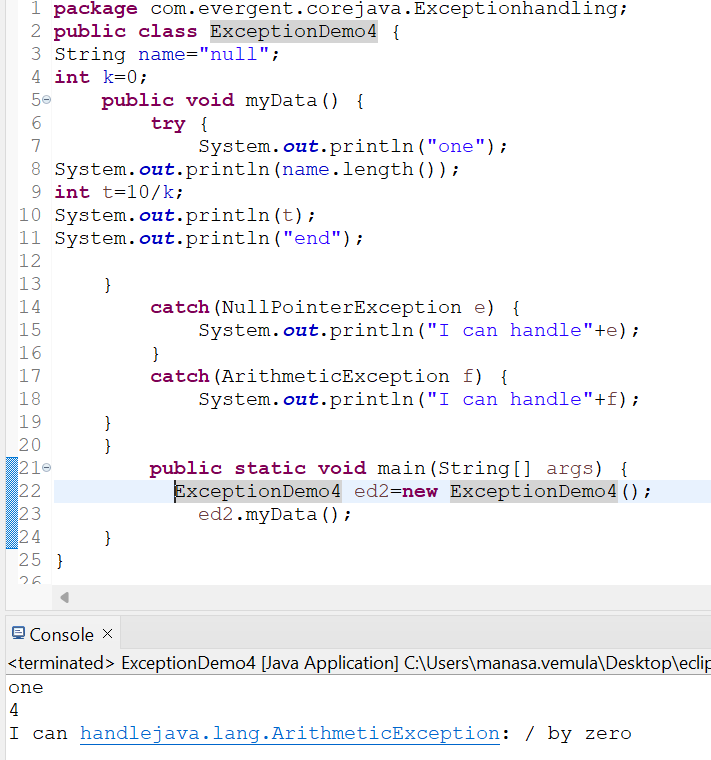
**Program 2:**



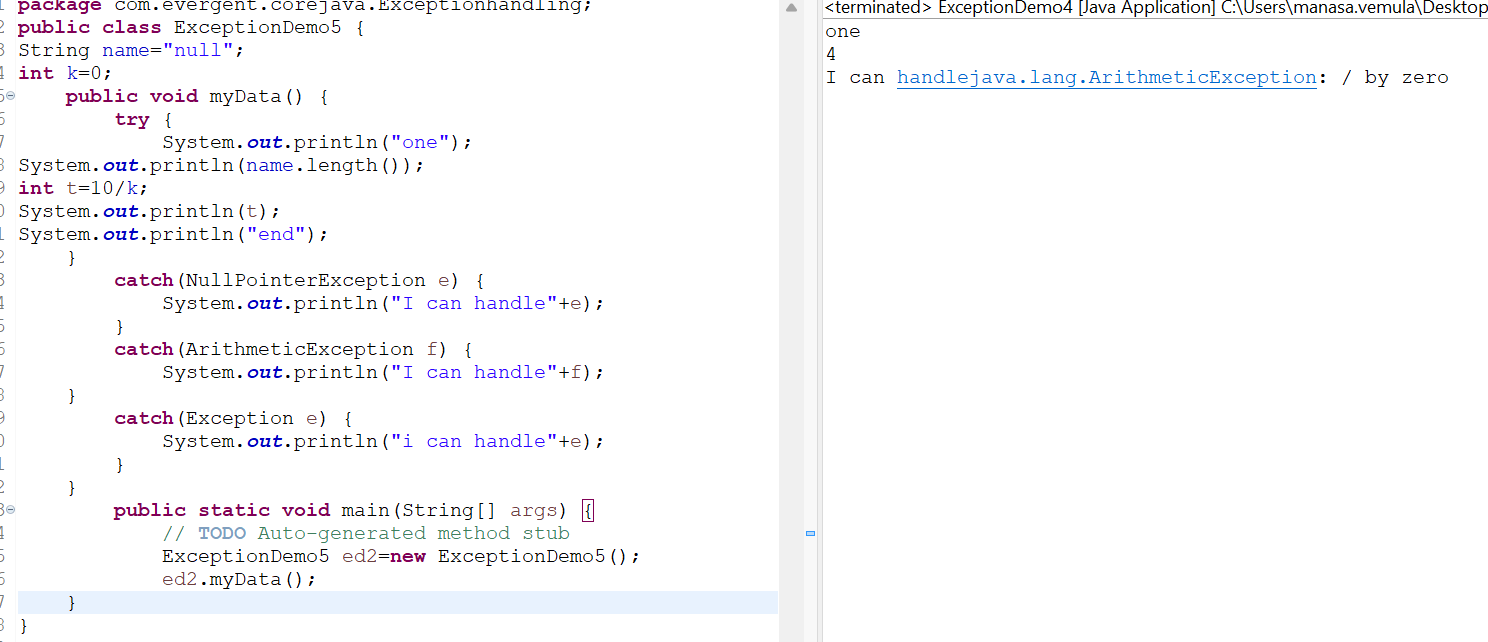
**Program 3:**



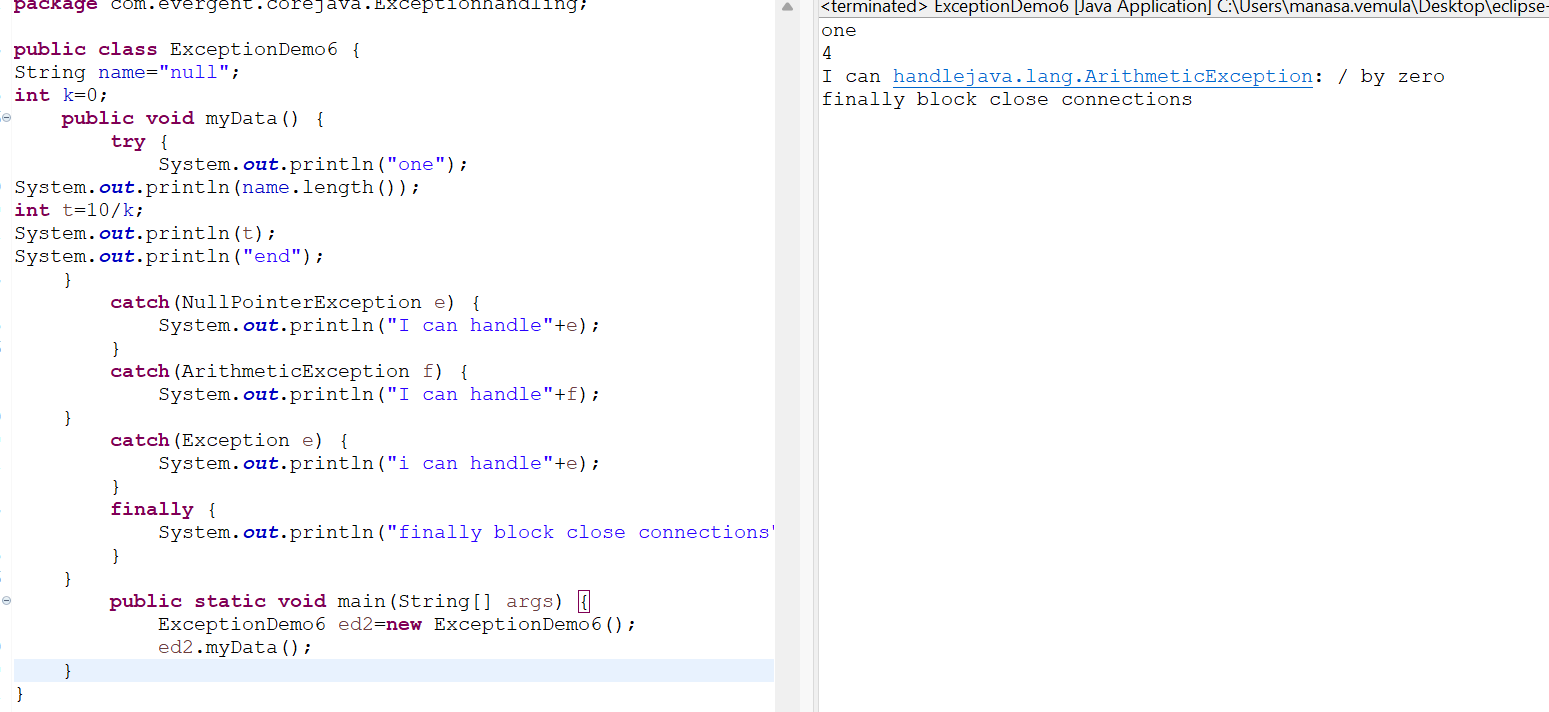
**Program 4:**



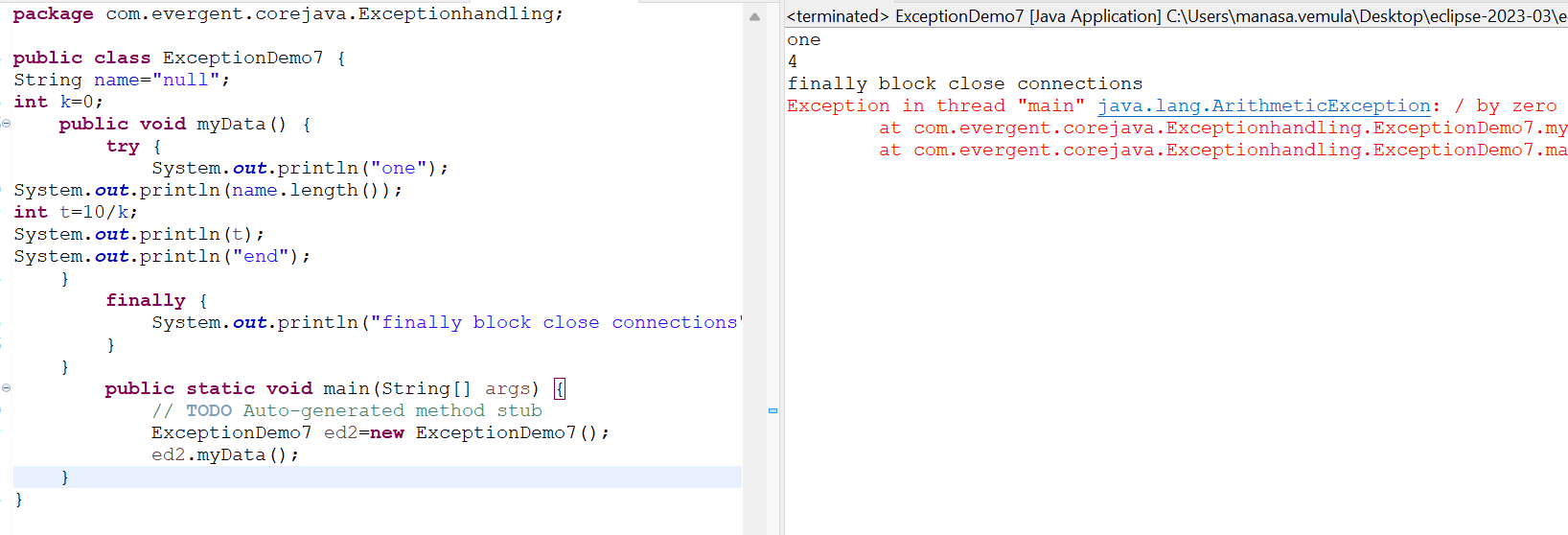
**Program 5:**



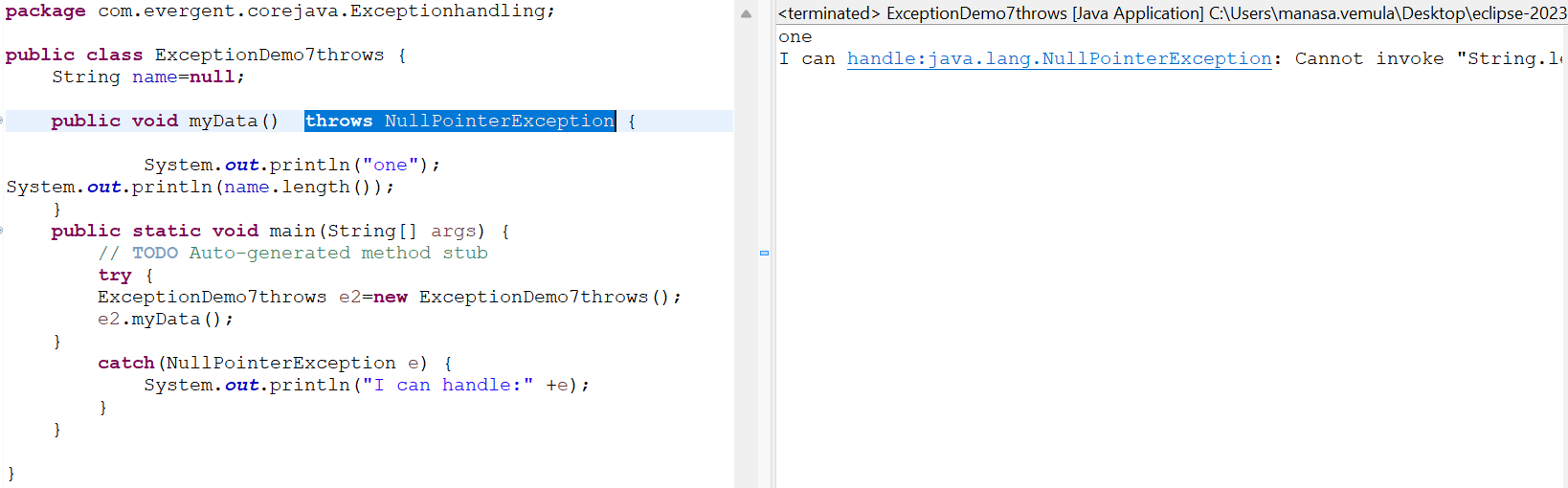
Program 6:



Program 7:



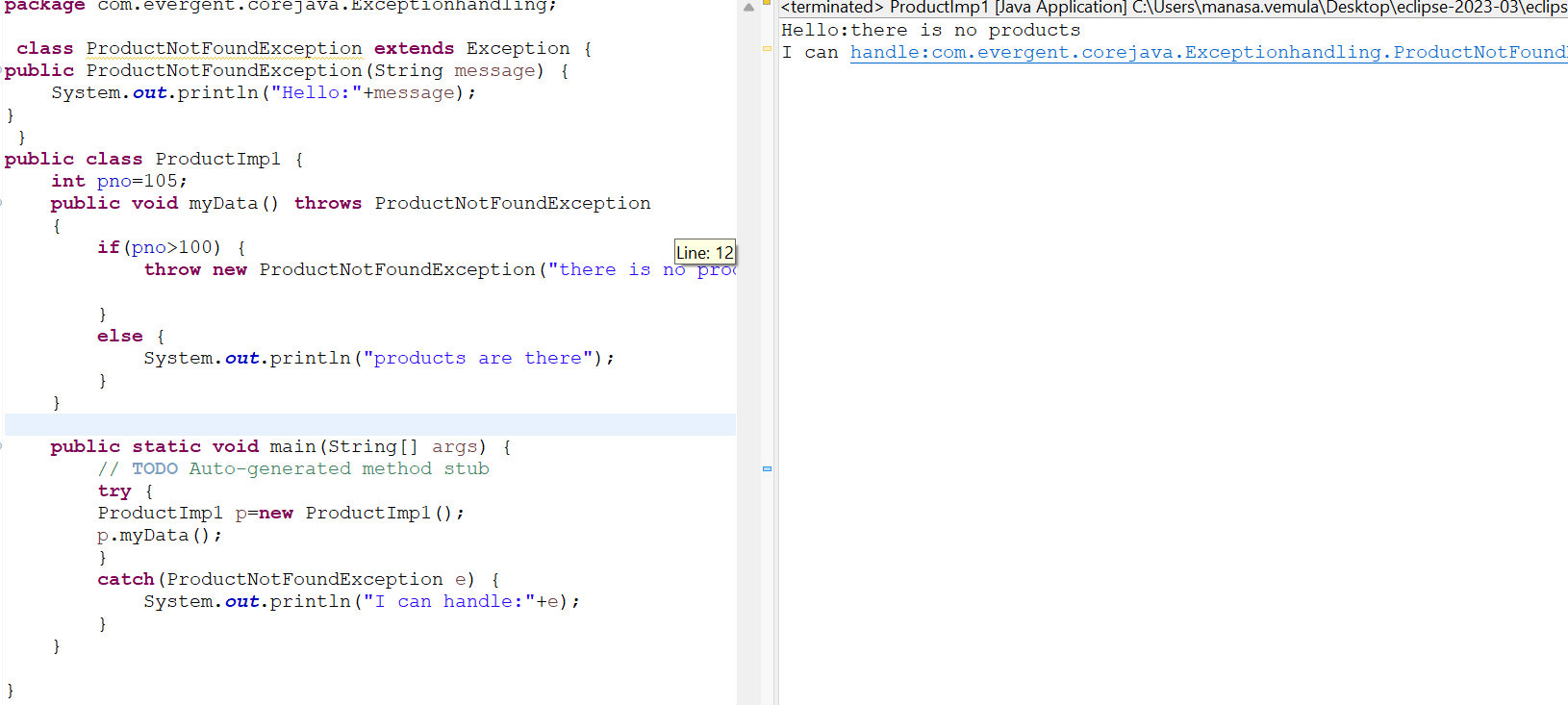
Program 8:



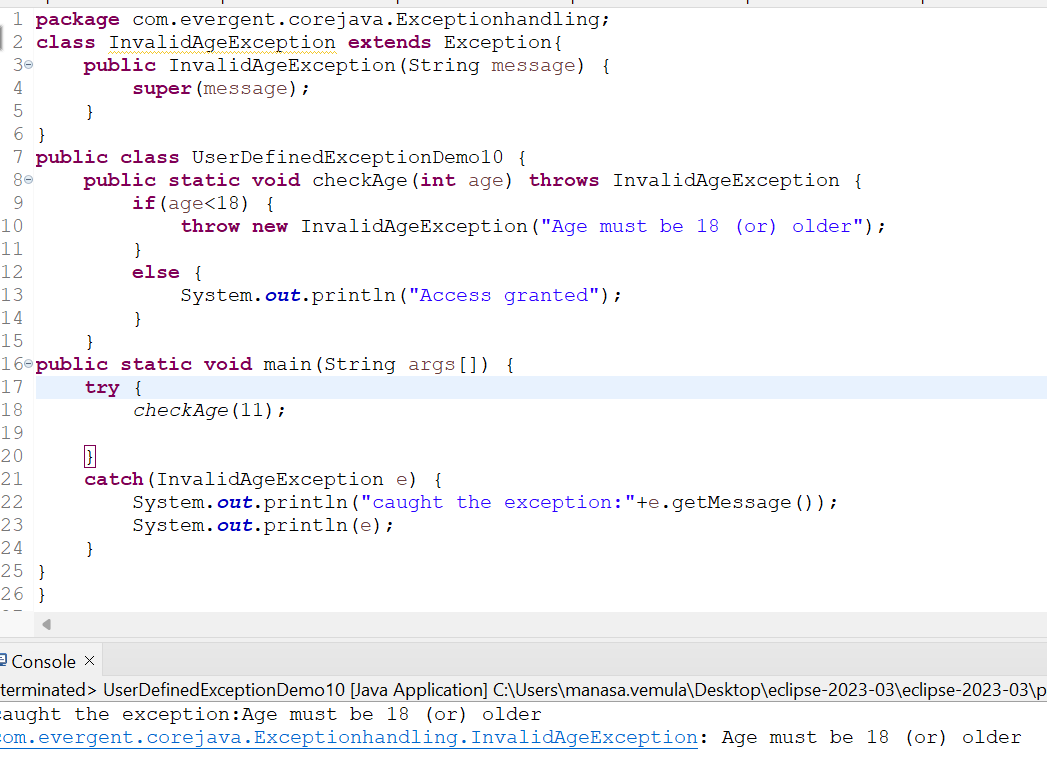
Program 9:



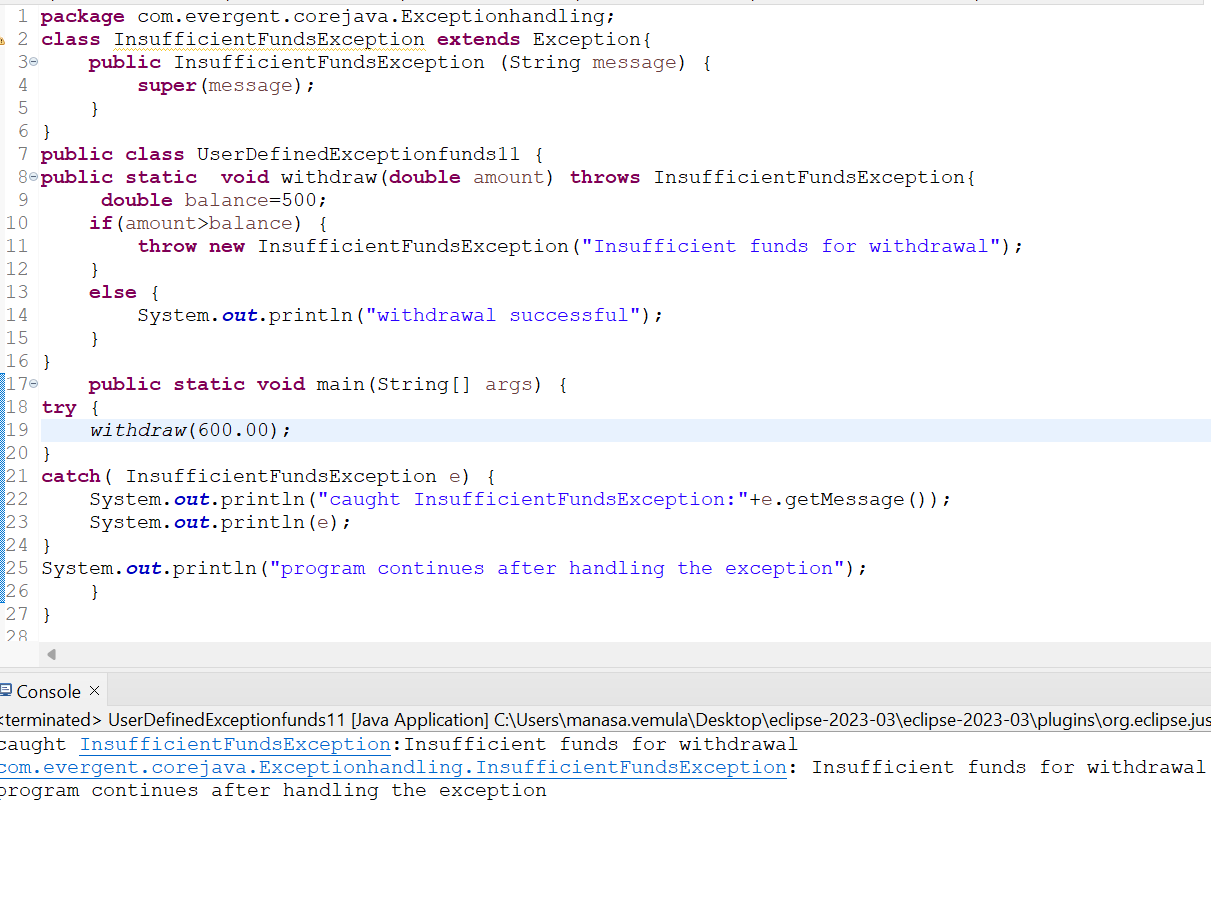
Program 10:



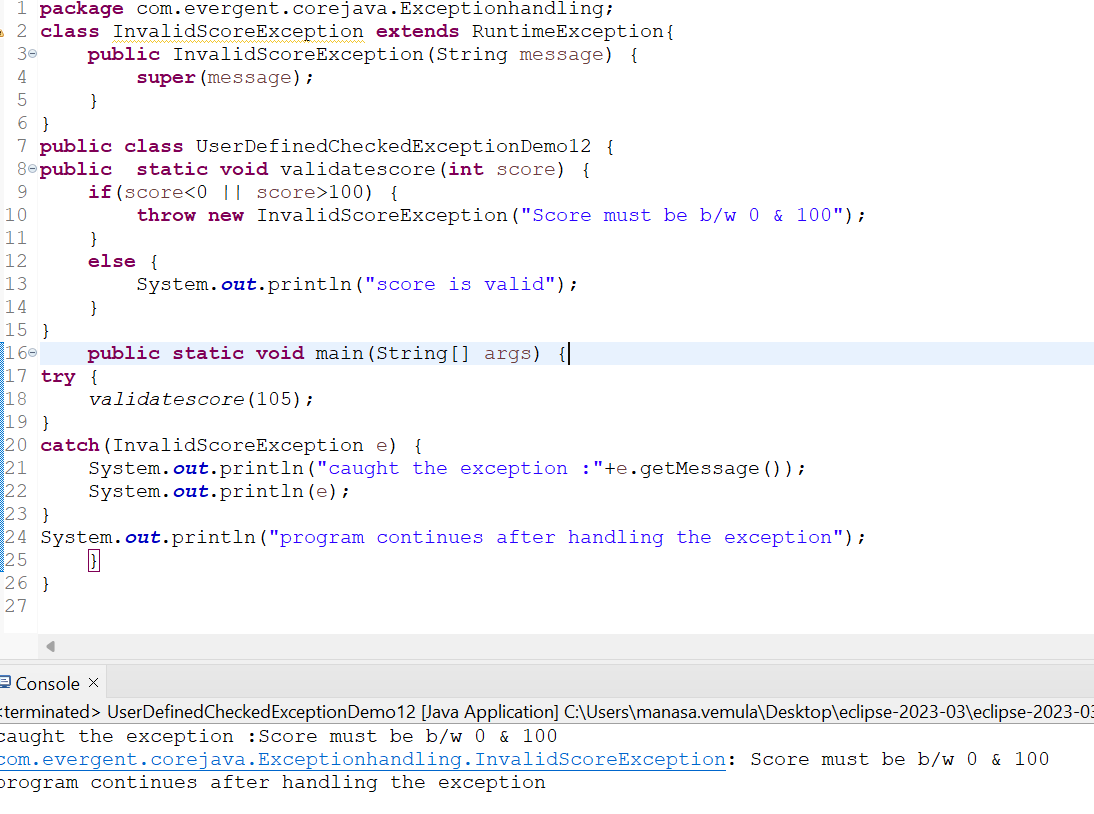
Program 11:



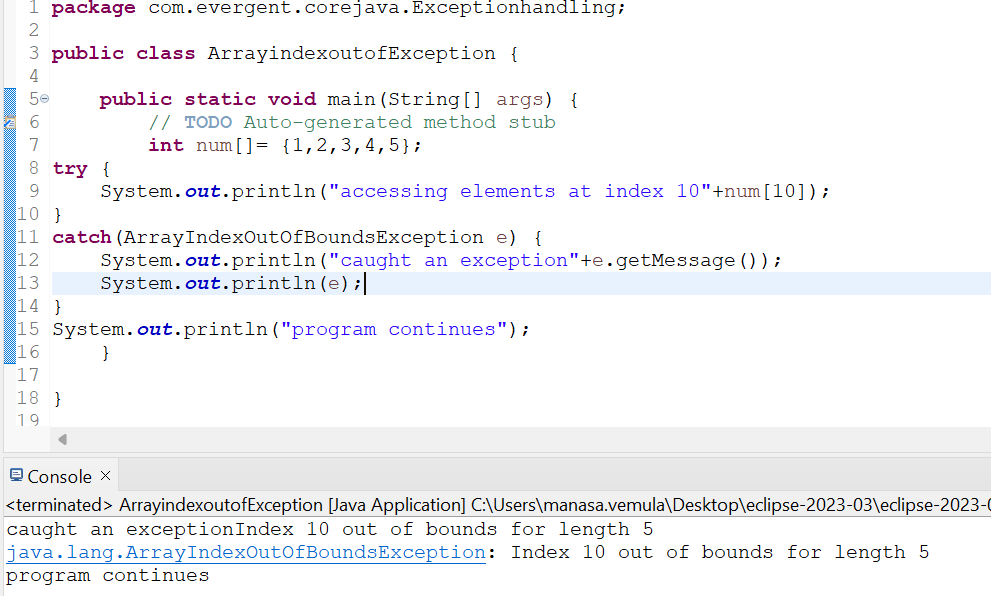
Program 12:



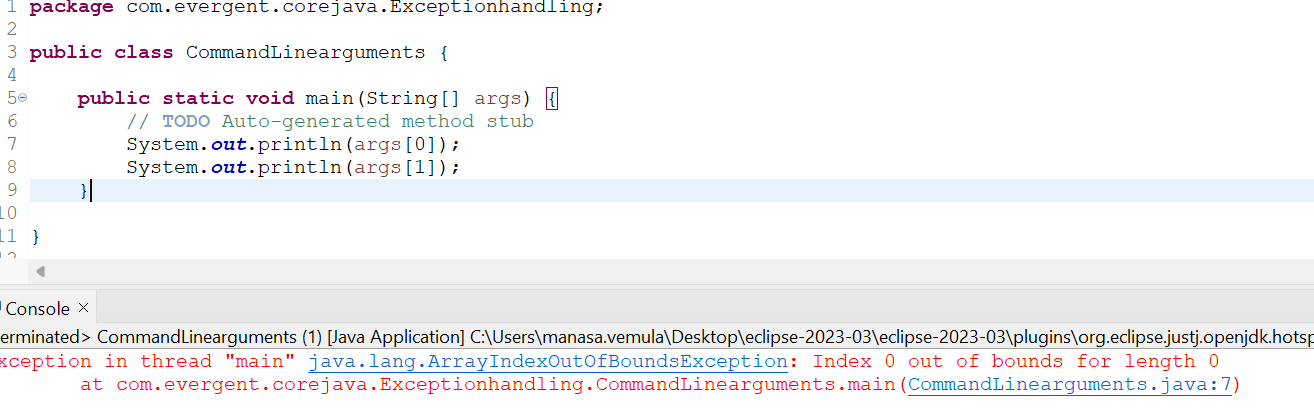
Program 13:



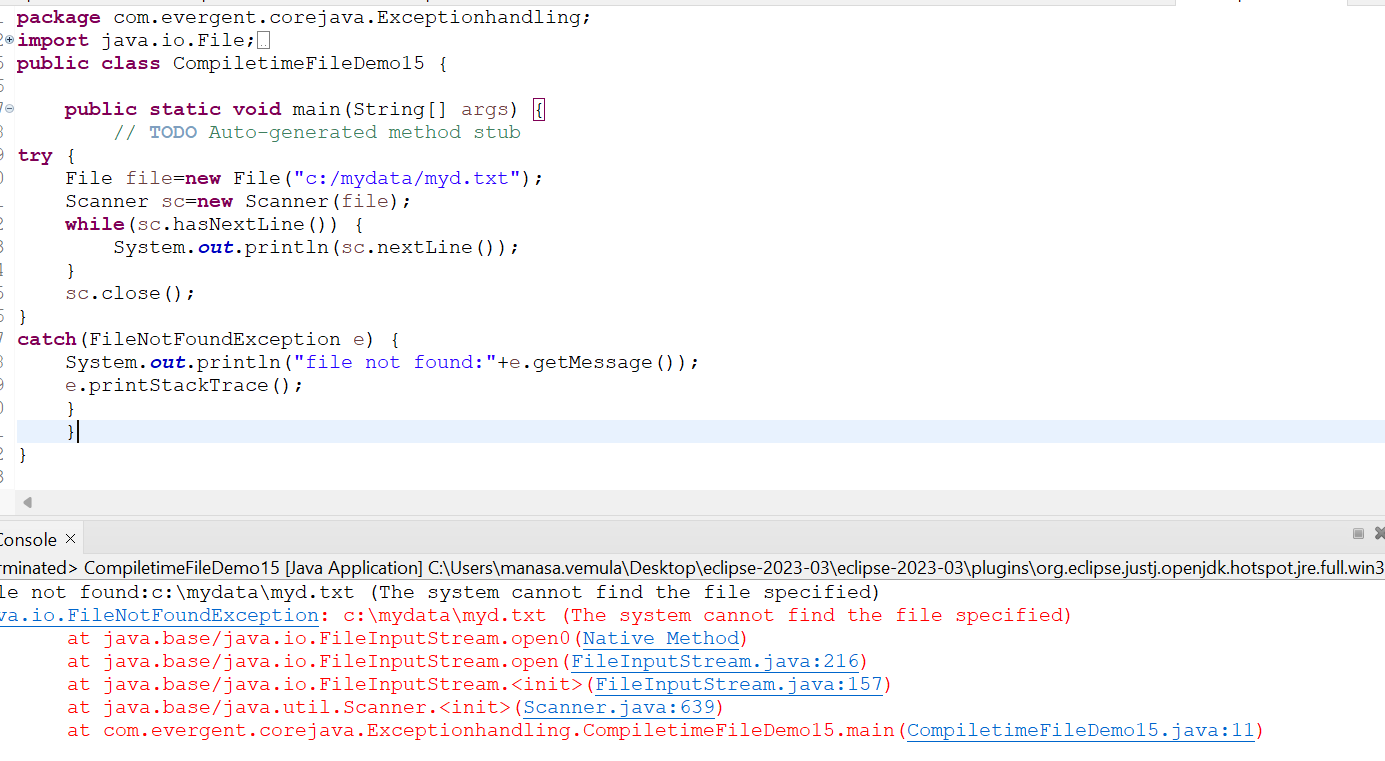
Program 14:



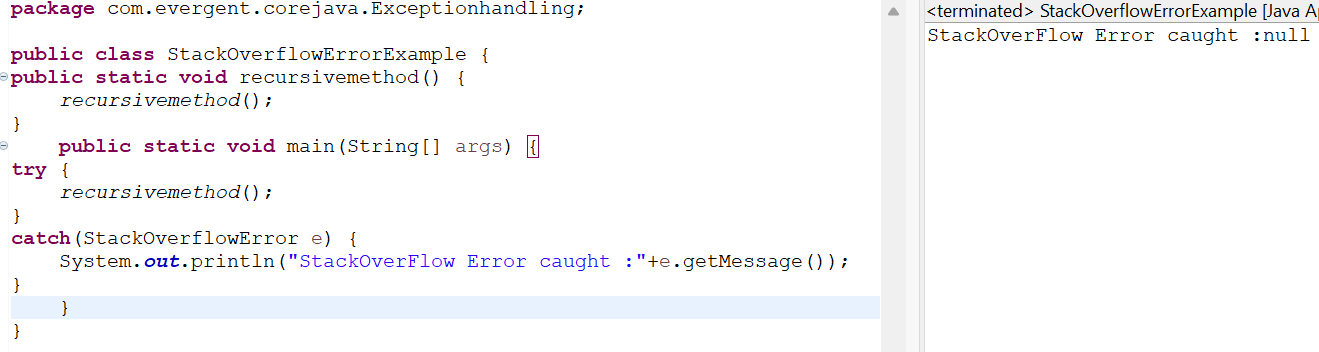
Program 15:



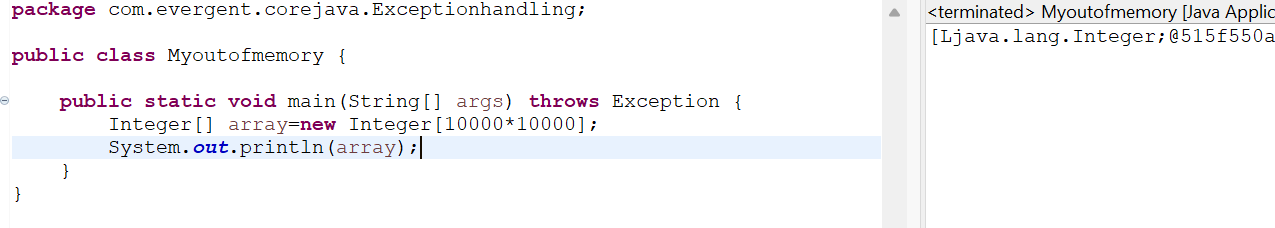
Program 16:



Program 17:



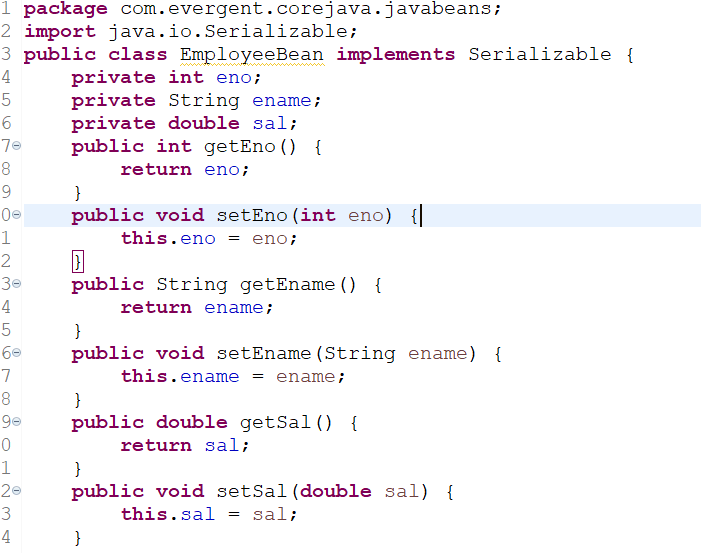
Program 18:

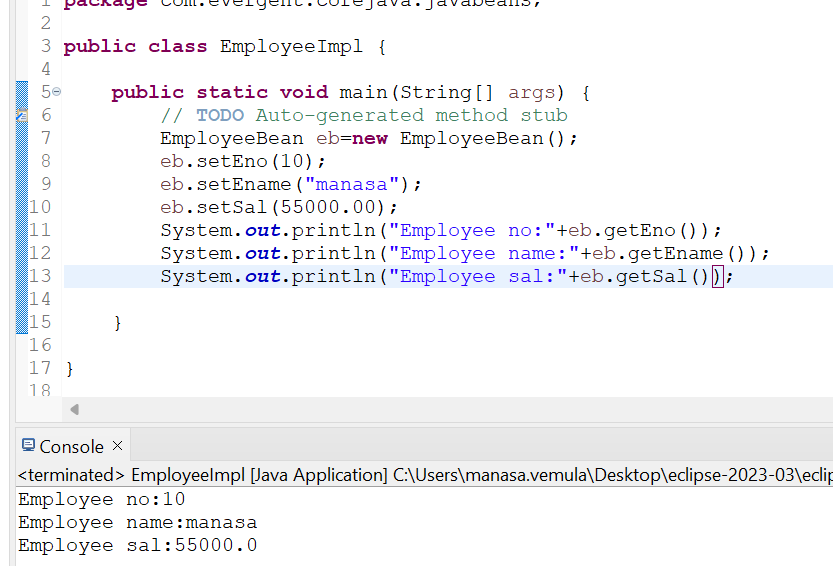


**Java Beans:**

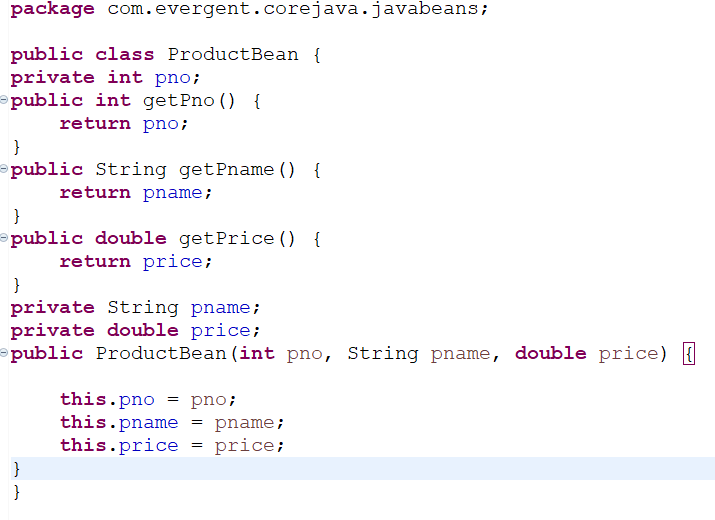
1. Java bean is a mechanism
2. Java bean is lightweight
3. All attributes are private and set/get methods are public
4. Implements java.io.serializable interface
5. We can achieve tightly encapsulation through java beans

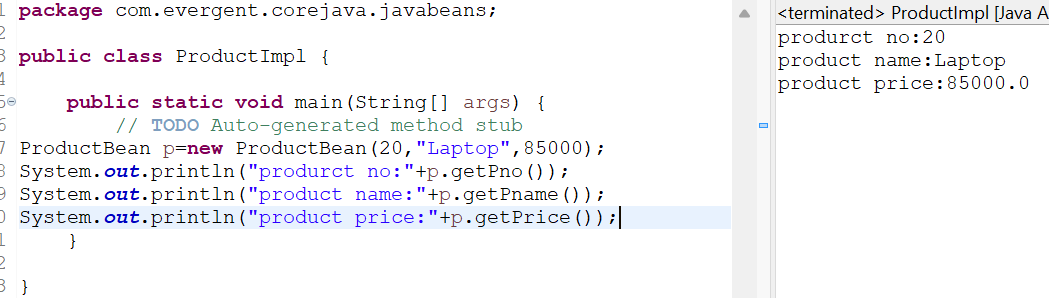
**Program 1:**



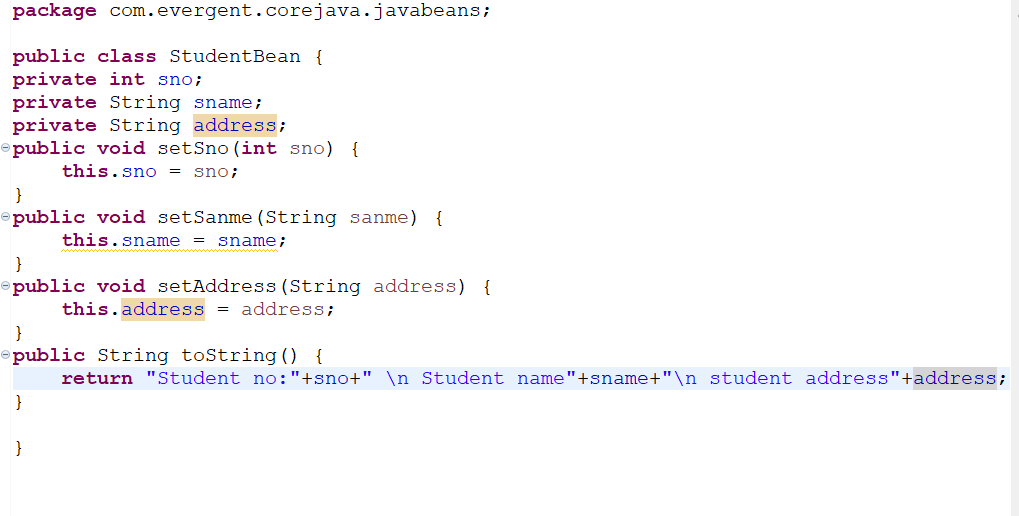


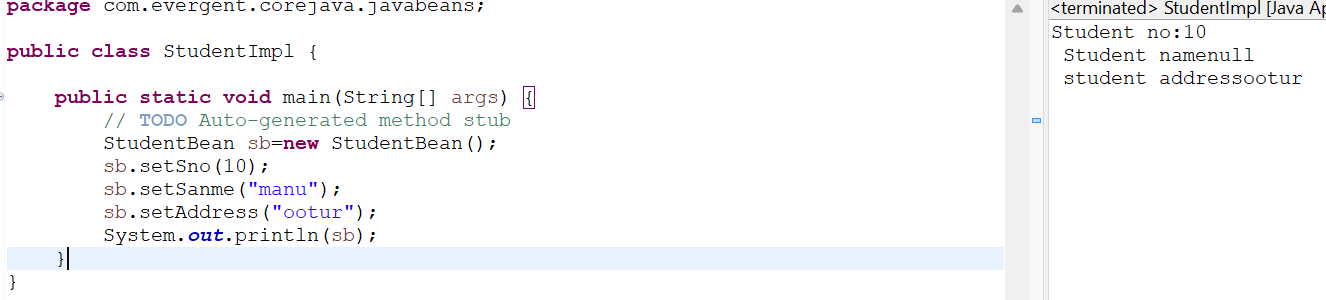
Program 2:





Program 3:



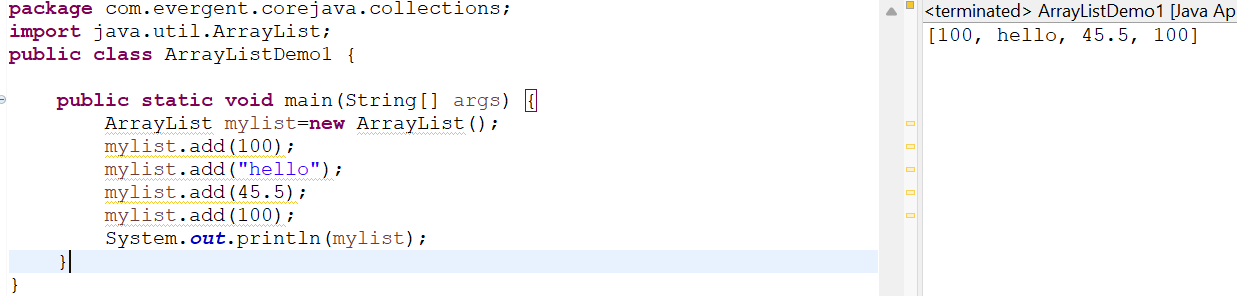


**Collections:**

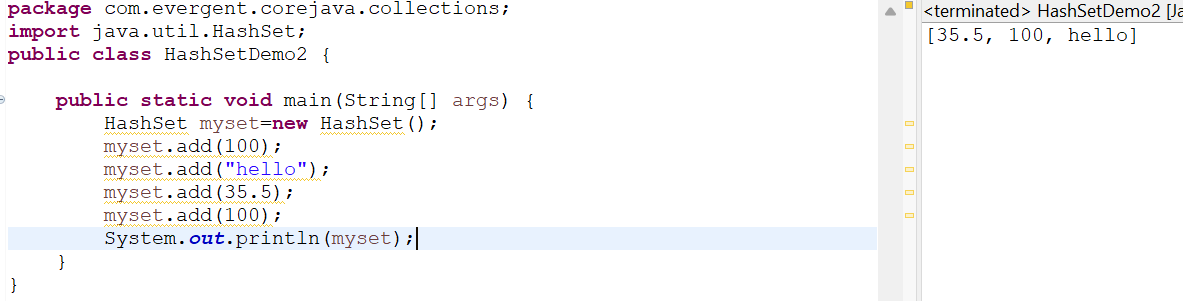
1.Set Items are stored randomly ,it will not allow duplicates.

2.List items are stored randomly ,it will allow duplicates.

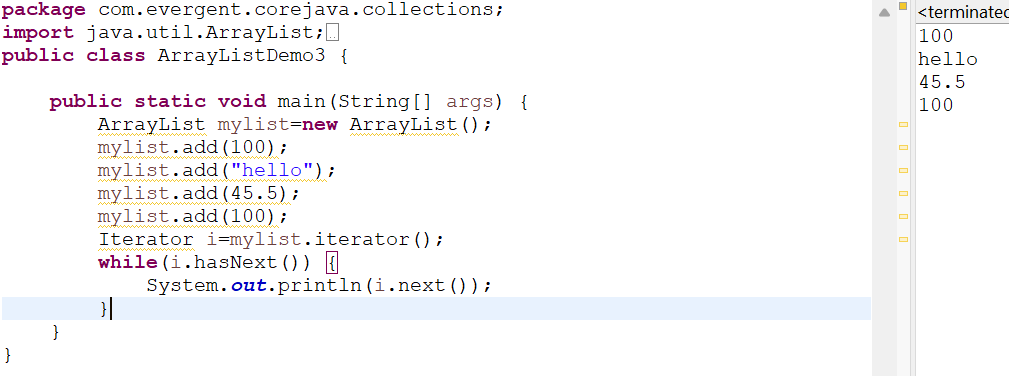
**Program 1:**



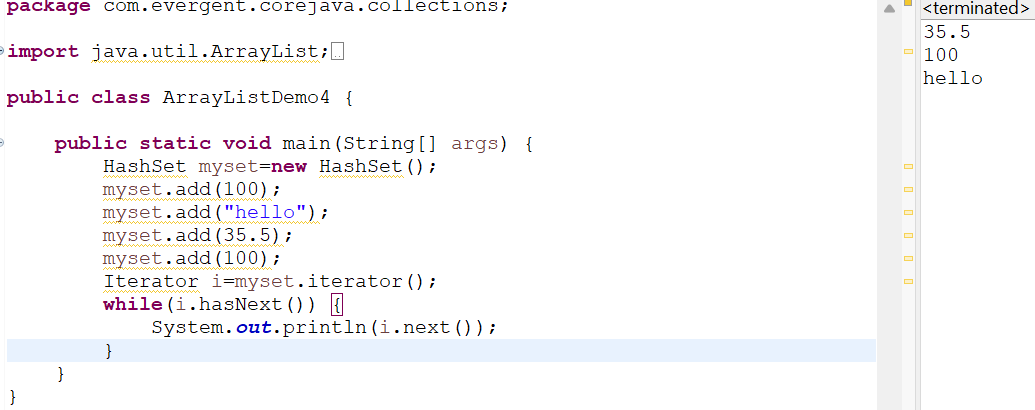
**Program 2:**



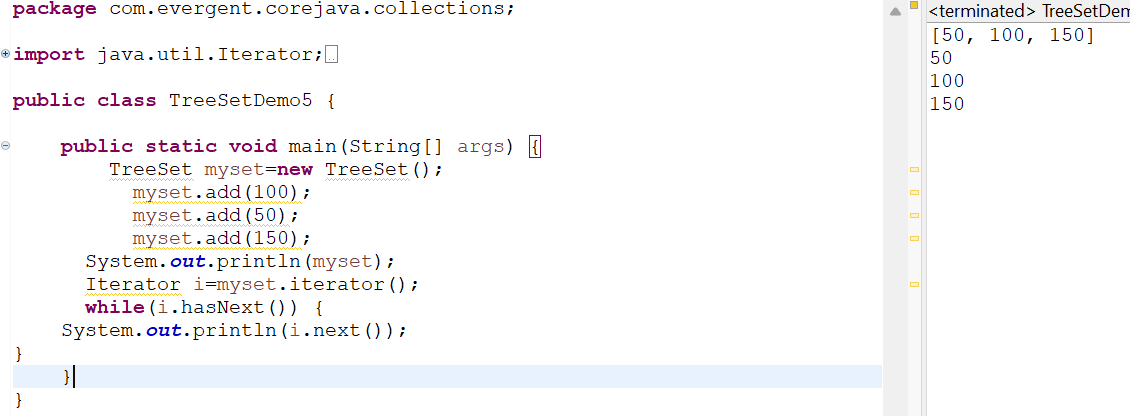
**Program 3:**



**Program 4:**

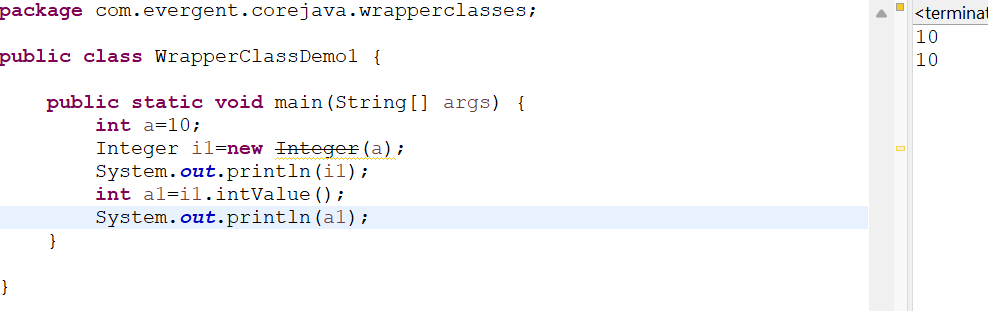


Program 5:

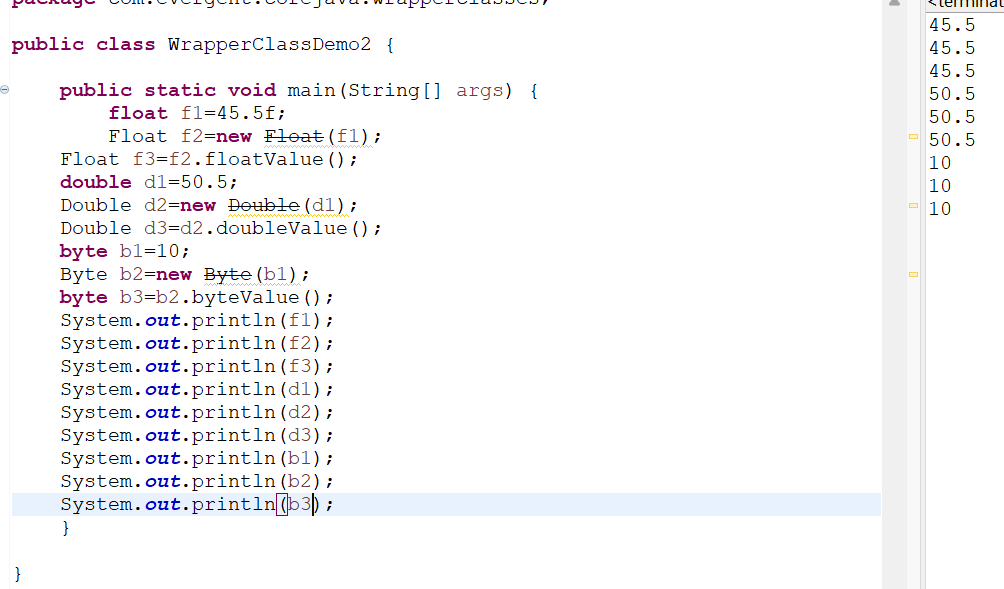


**Wrapper Classes:**

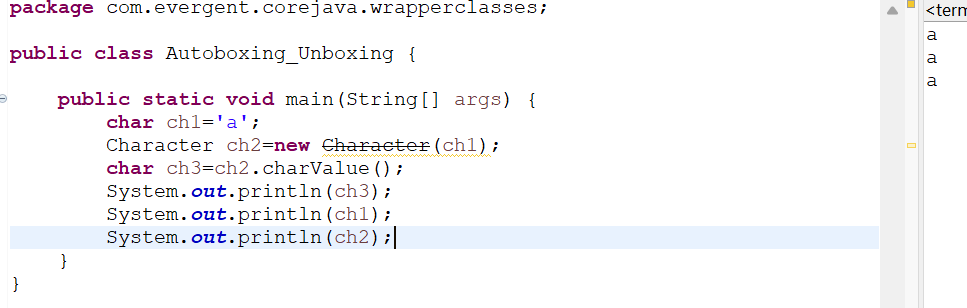
**Program 1:**



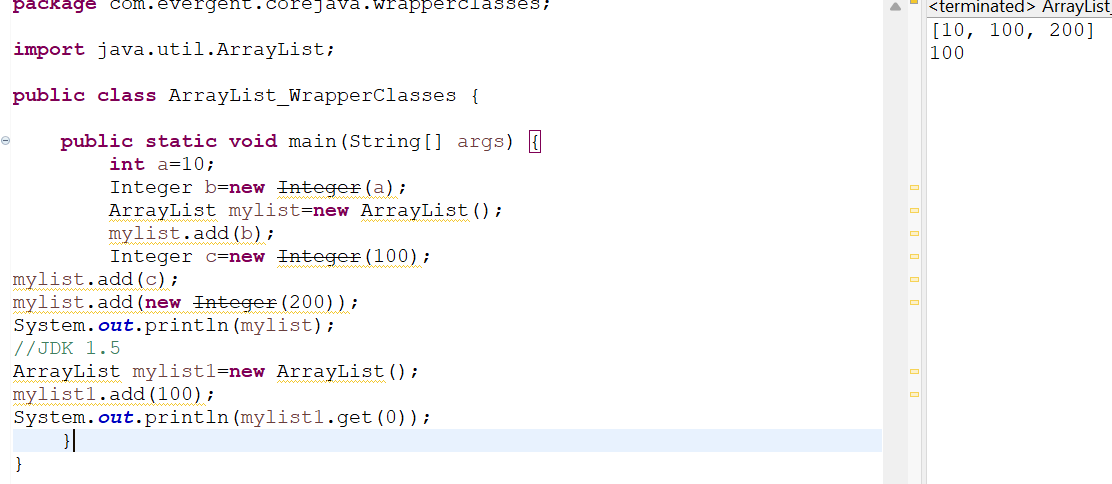
**Program 2:**



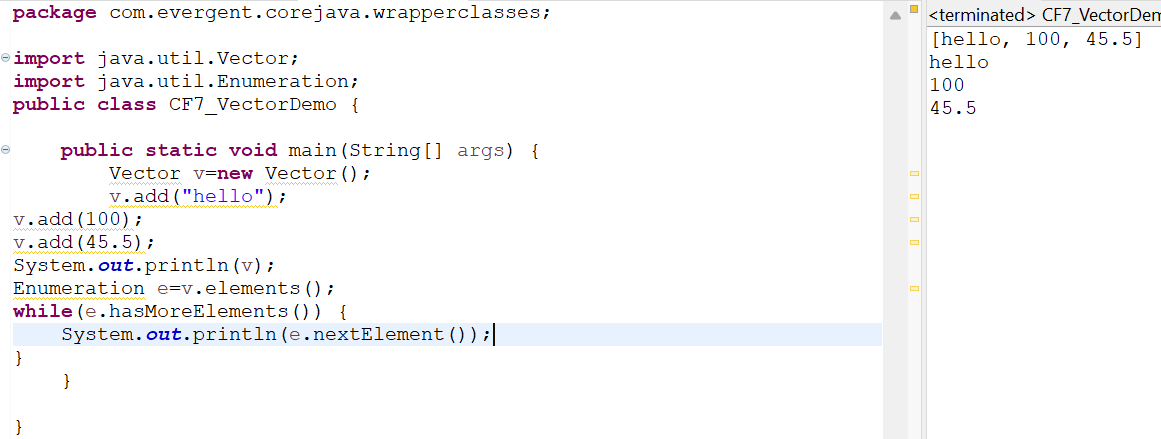
**Program 3:**



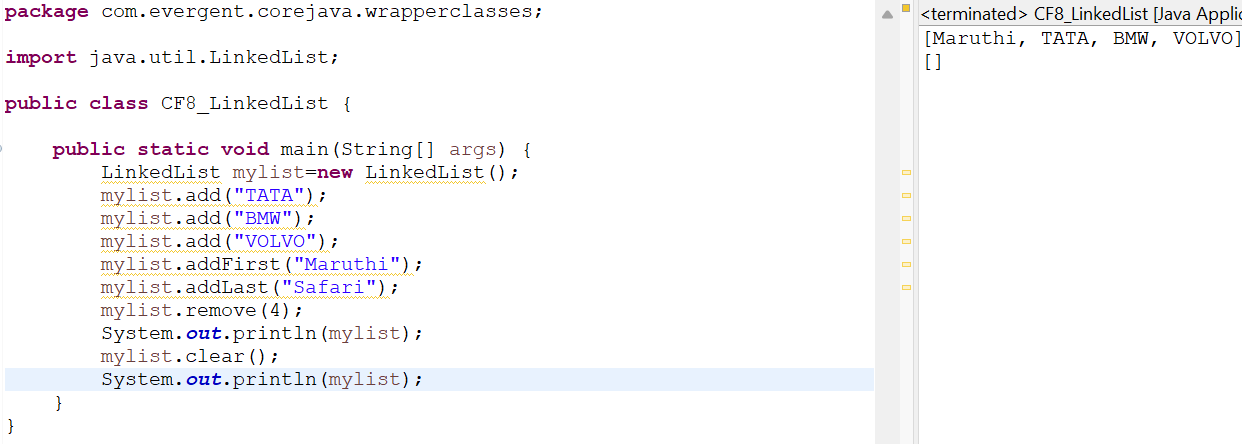
**Program 4:**



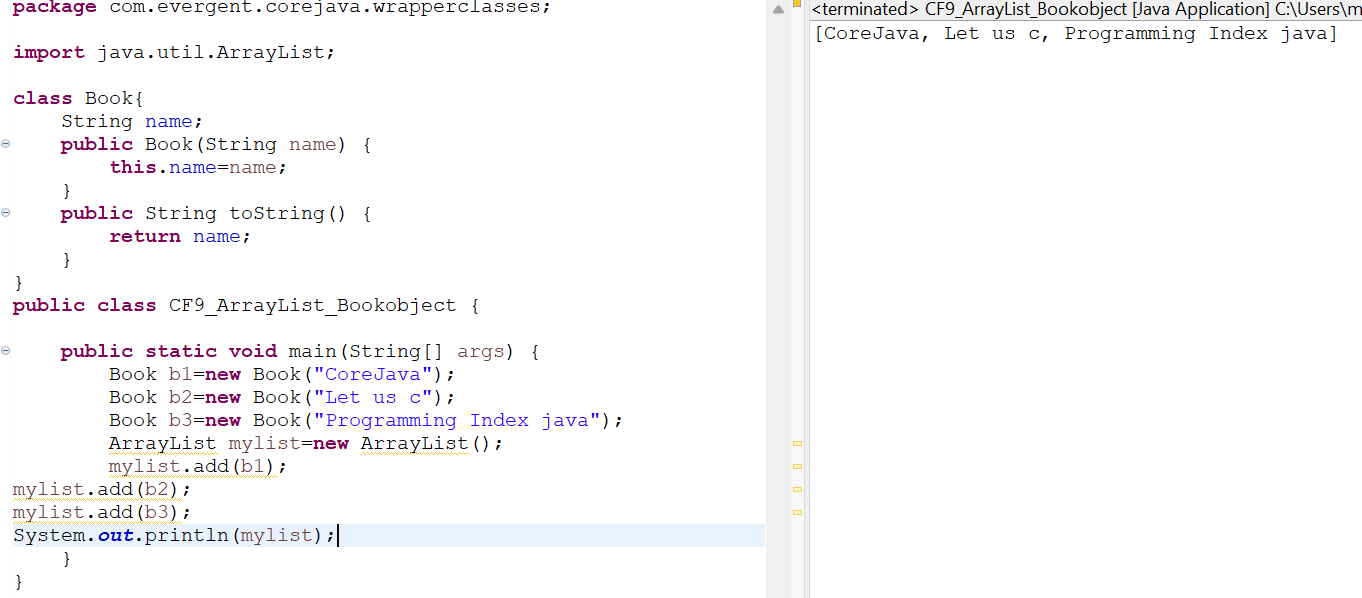
**Program 5:**



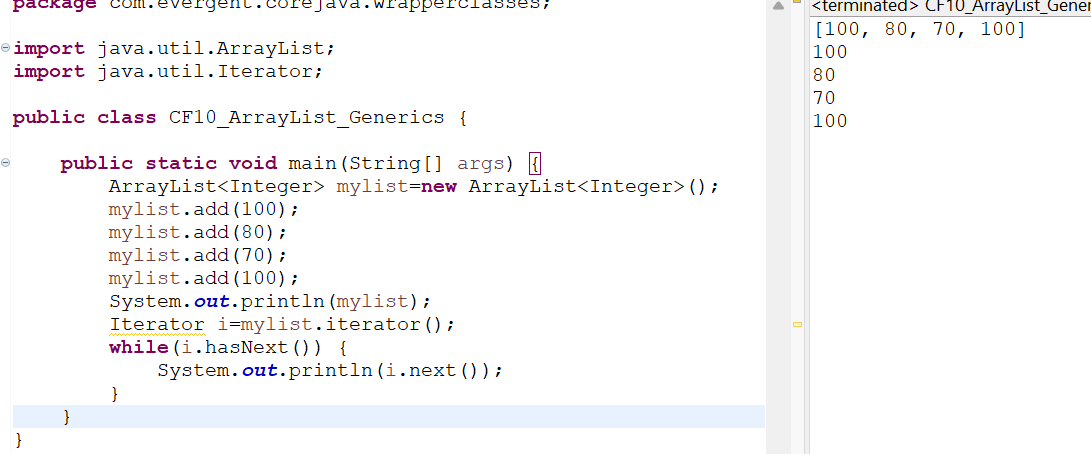
**Program 6:**



**Program 7:**



Program 8:



Program 9:

