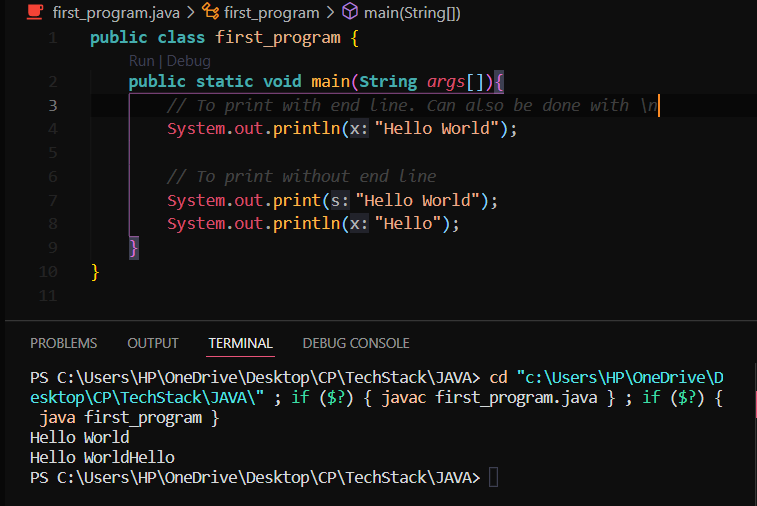
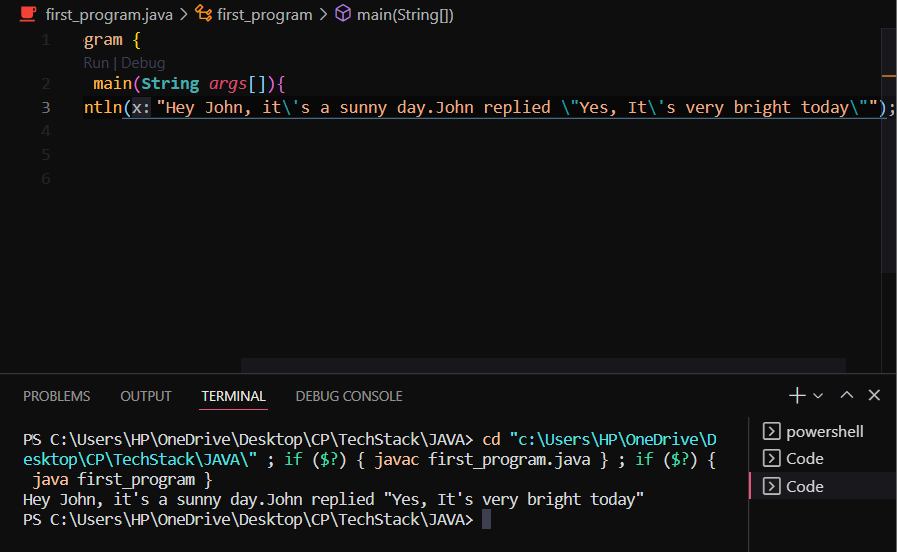
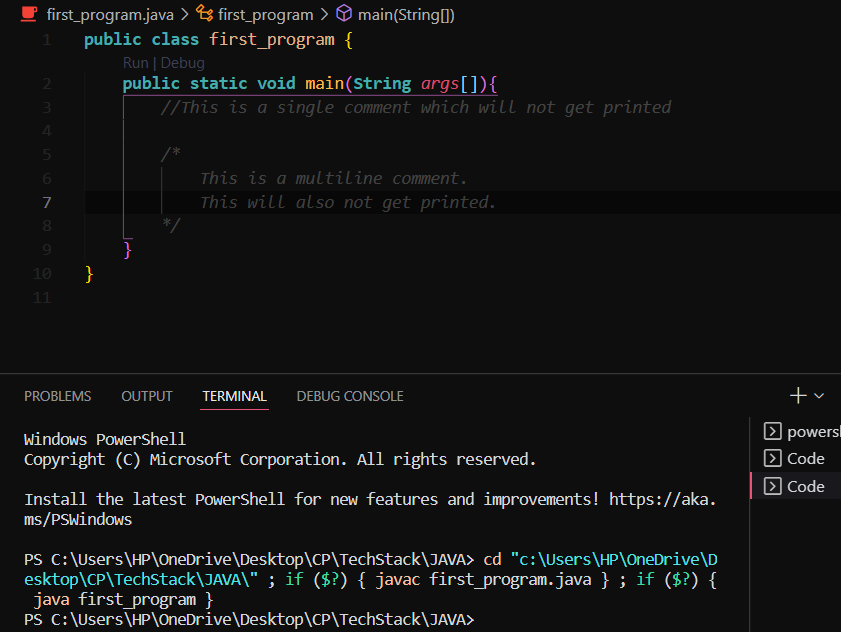
1. First Program



To print the text with quotes use \ in front of quotes.



**Comments**

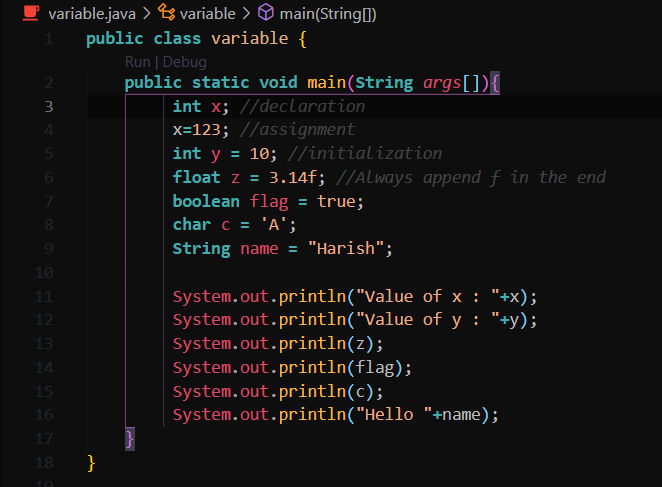
****

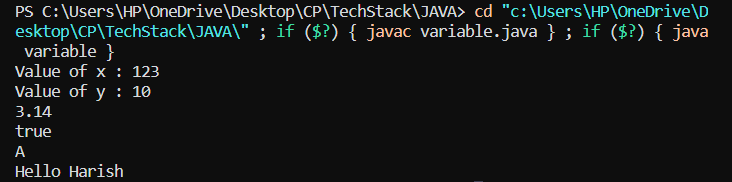
1. Variables

Used to store some values

Types:

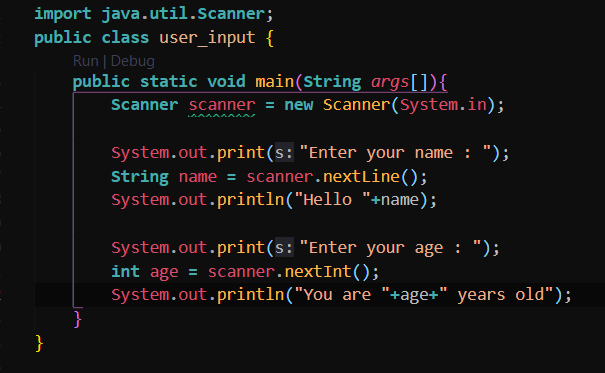
1. boolean
2. byte
3. short
4. int
5. long
6. float
7. double
8. char
9. String

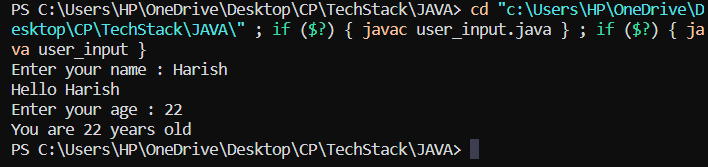




1. User Input

Done using Scanner class

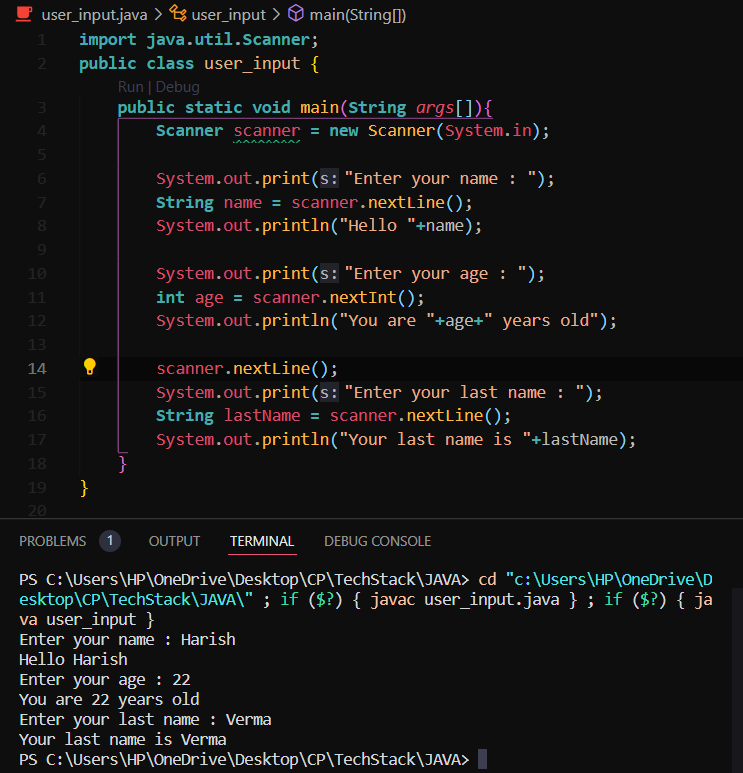




nextLine() 🡪 String

nextInt() 🡪 Integer

Note : Always add a scanner.nextLine() statement after taking input an integer.

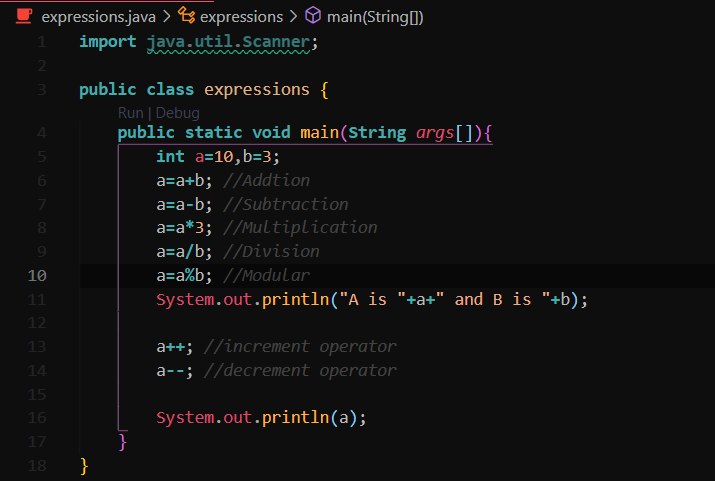


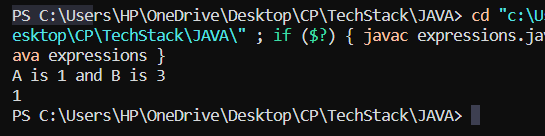
1. Expressions

Expressions = operands & operators

Operands = values, variables, numbers, quantity

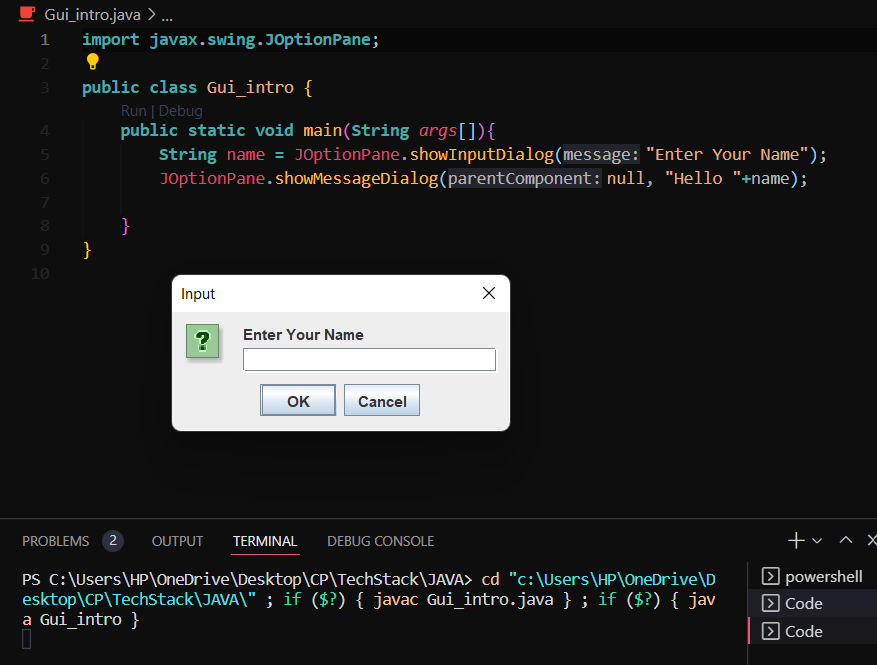
Operators = +, -, \*, /, %

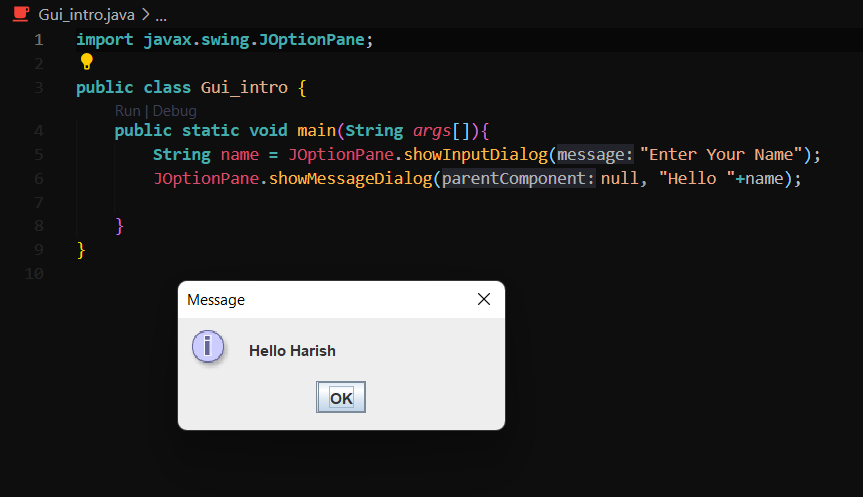




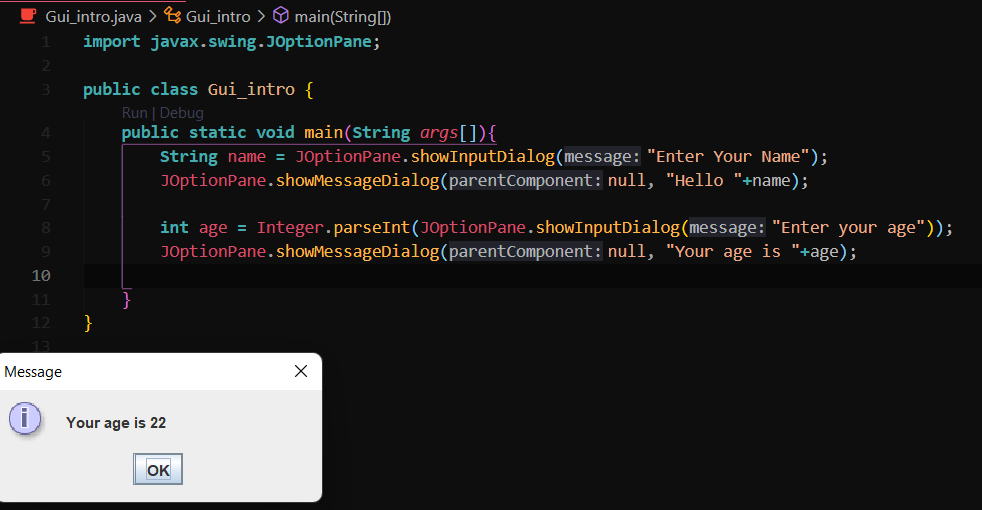
1. GUI Intro

JOptionPane is used.

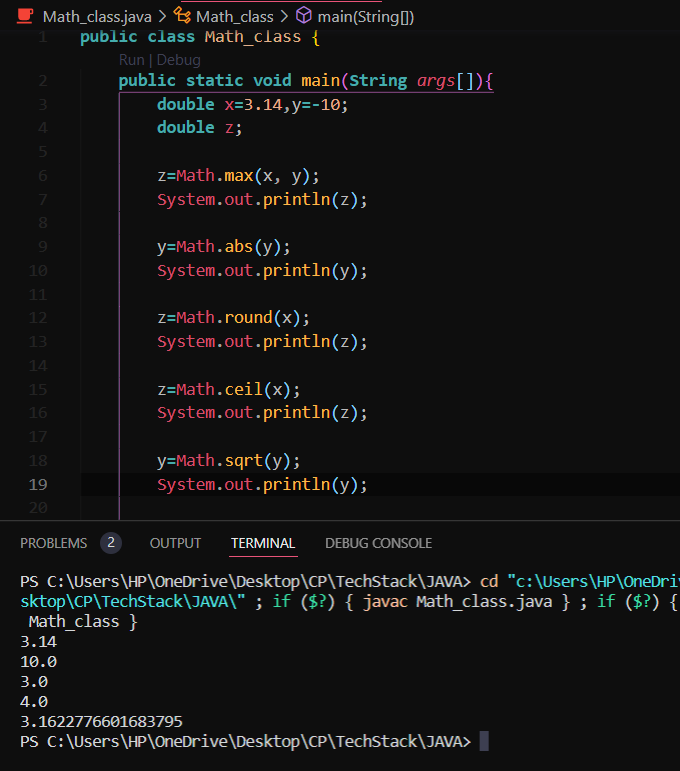




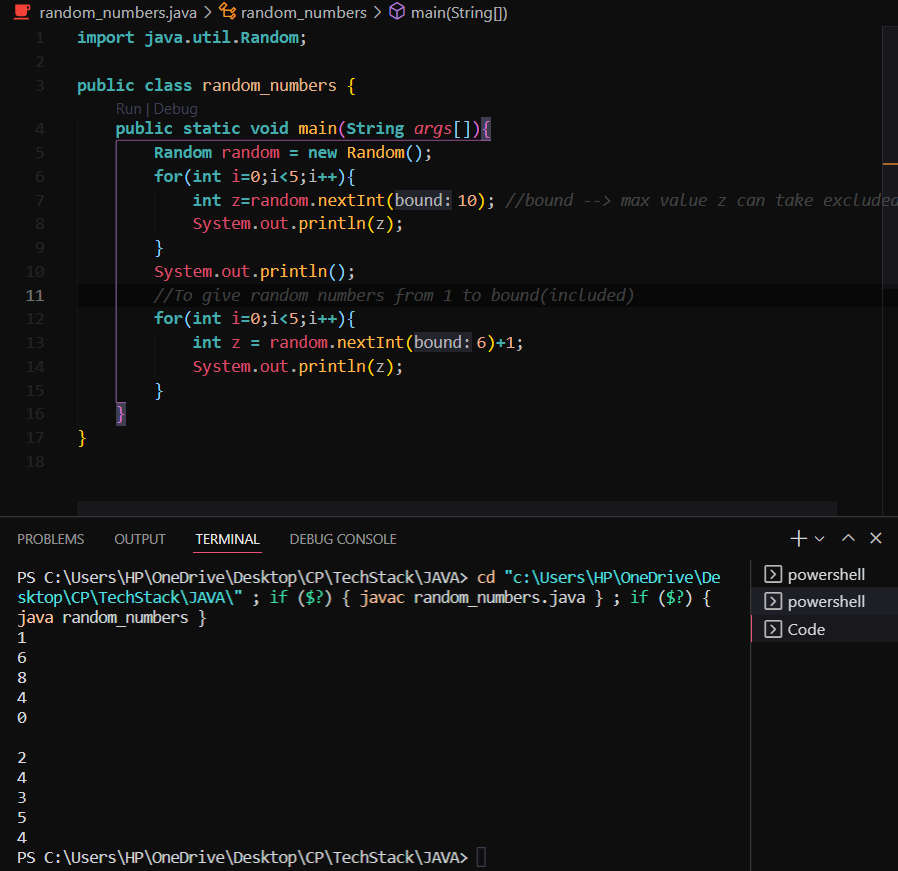
To convert string into integer or any other, use respective Parse method.



1. Math class

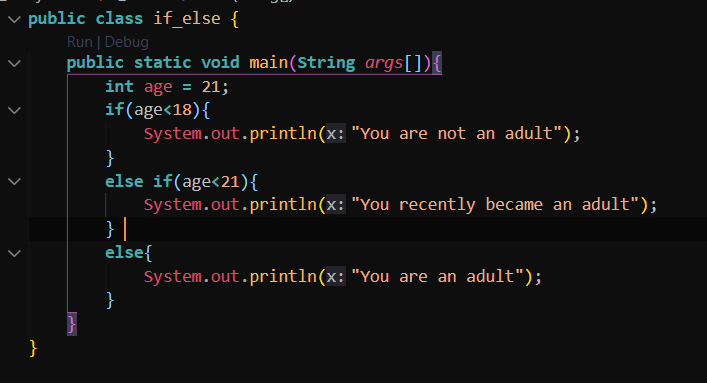


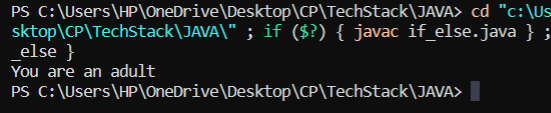
1. Random Numbers

****

1. If-else\_if-else statements

Performs a block of statement if the condition evaluates to true



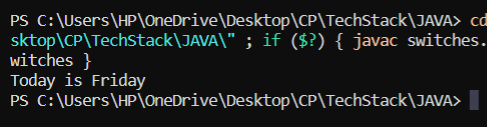


You can have infinite else if statements.

1. Switch

Used to eliminate multiple else if statements





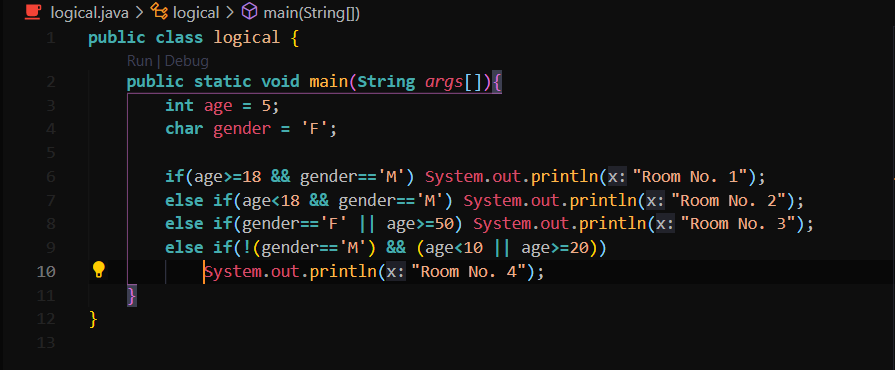
Note : Use break else it will print all blocks once it evaluates to true.

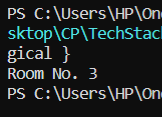
1. Logical Operators

To test multiple conditions

Types:

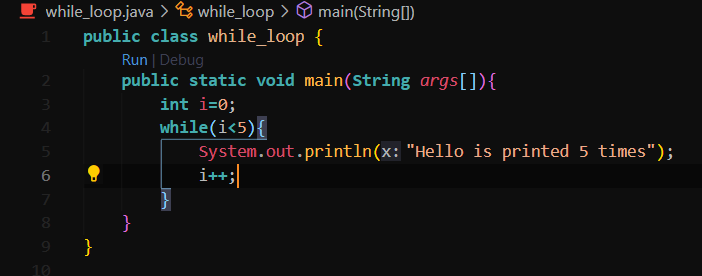
1. AND (&&) 🡪 When all conditions are true, it is true
2. OR (||) 🡪 When atleast one condition is true, it is true
3. NOT (!) 🡪 Changes false to true, and vice versa

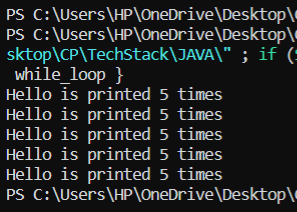




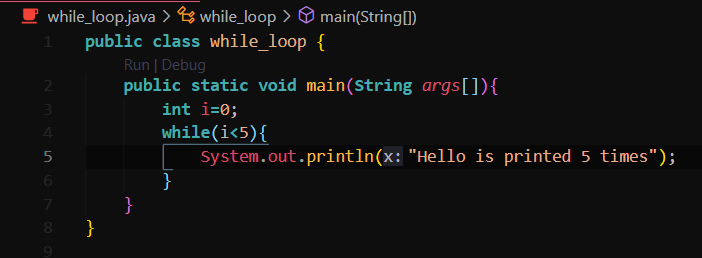
1. While Loop

Statements are executed till the condition is true





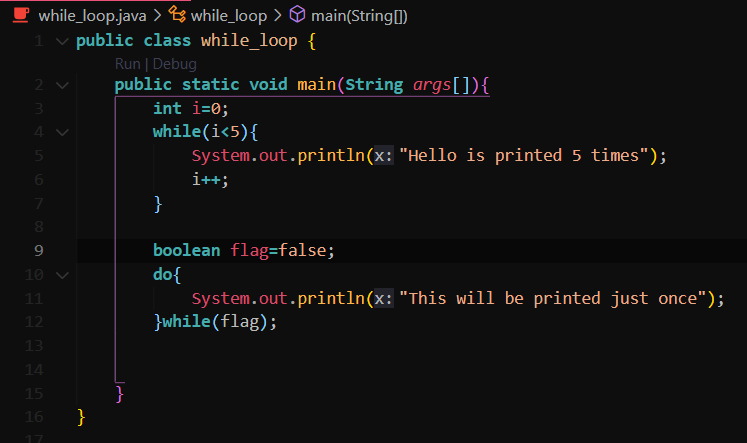
If i is not incemented it may result in infinite loop.

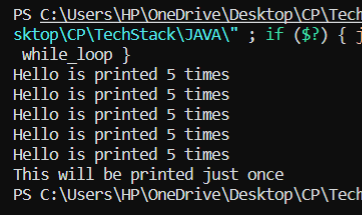


Output : Infite time print statement

**do-while loop**

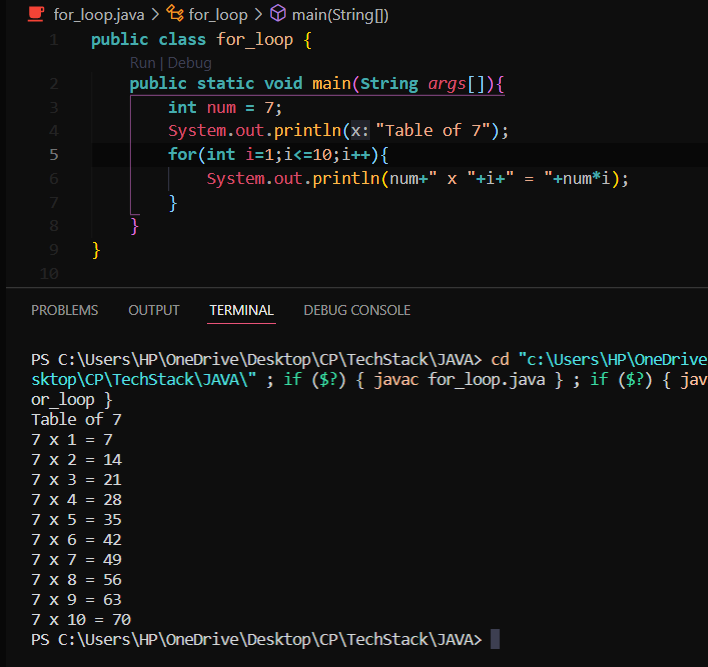
same as while loop, but it executes atleast one even if the condition is false.





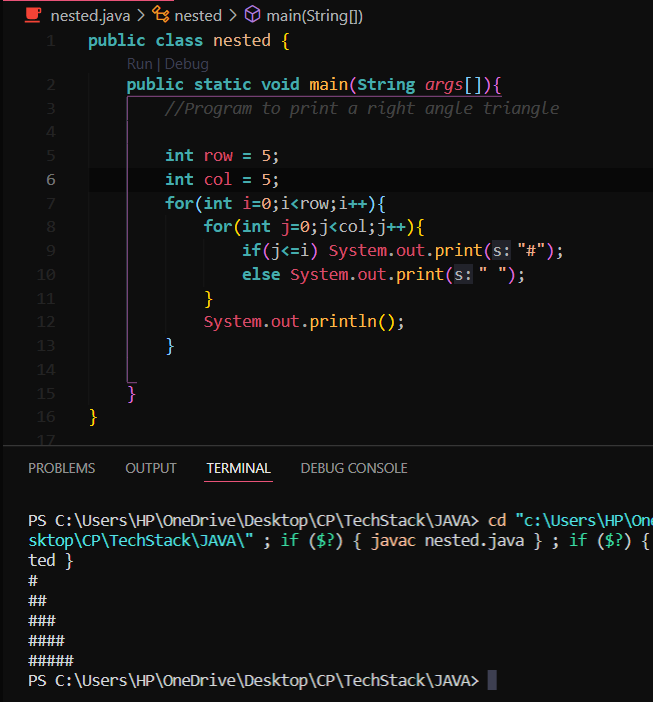
1. For loop

Used when amount of iterations are known



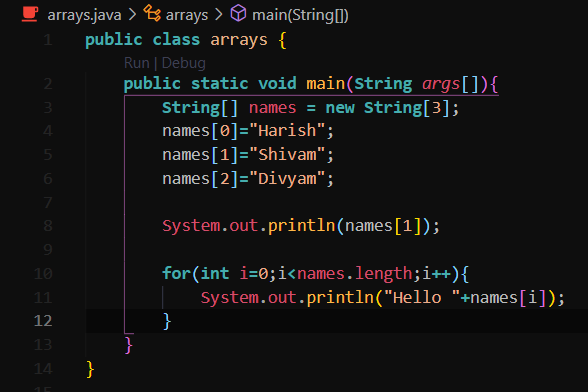
1. Nested Loops

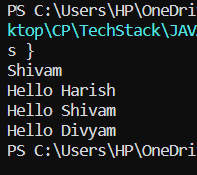
Loops within loops



1. Array

Used to store the objects of same data type.



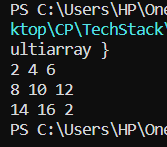


Use .length to get the size of the array

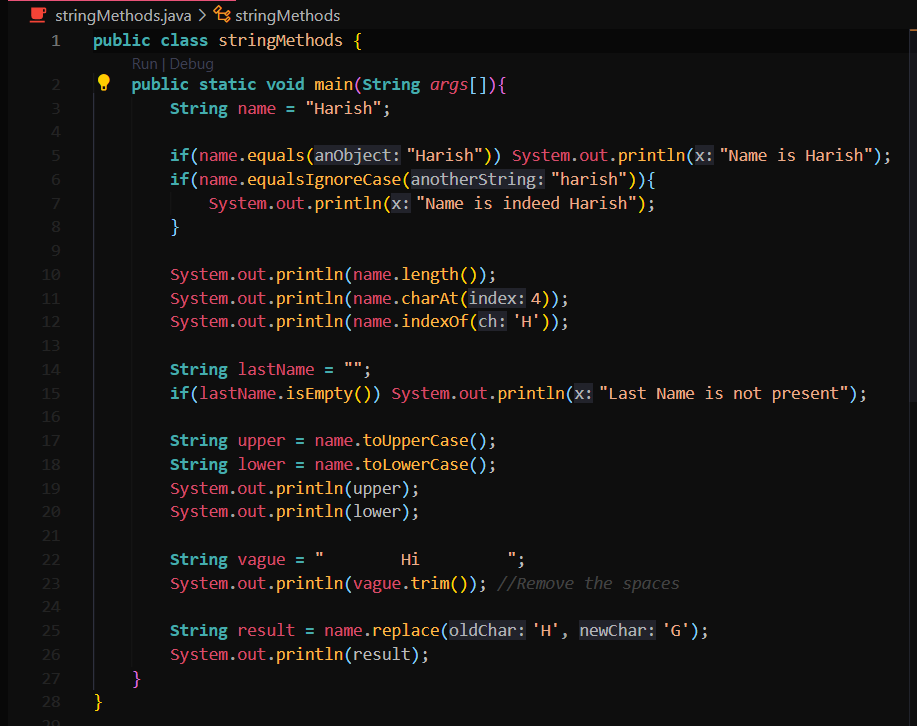
1. 2D array

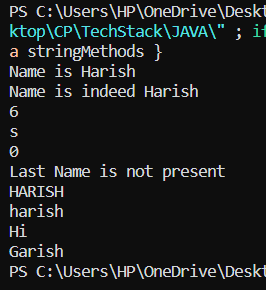
Array within array





1. String methods





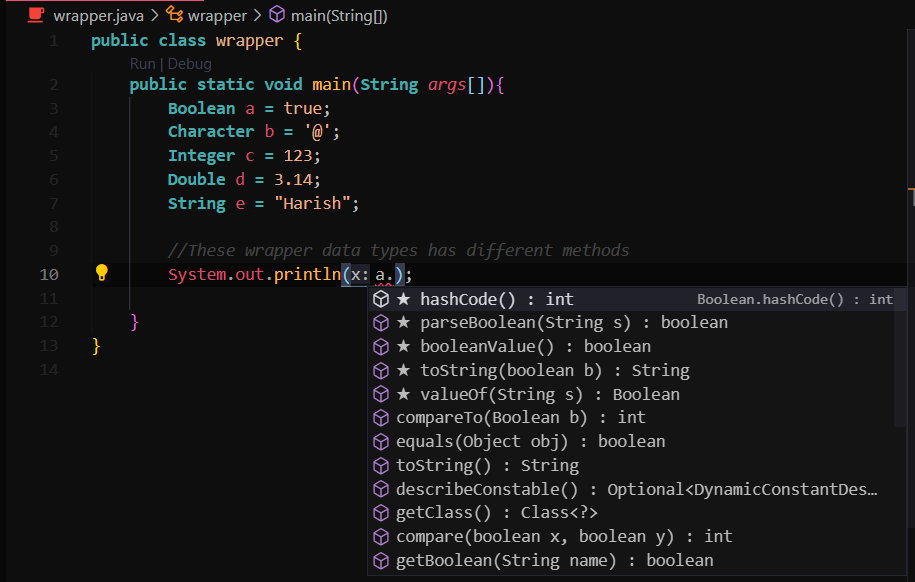
1. Wrapper Class

Primitive 🡪 boolean, char, int, double

Wrapper 🡪 Boolean, Char, Int, Double

Autoboxing 🡪 Automatic conversion that the Java compiler makes between primitive and wrapper

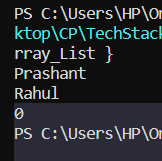
Unboxing 🡪 The reverse of autoboxing. Automatic conversion of wrapper class to primitive



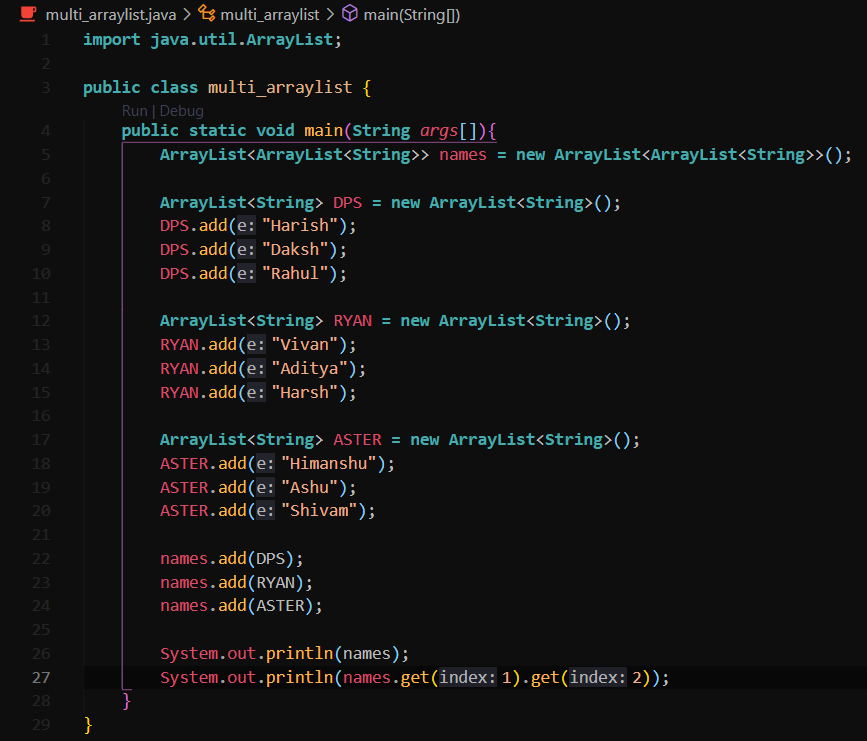
1. ArrayList

Dynamic Array

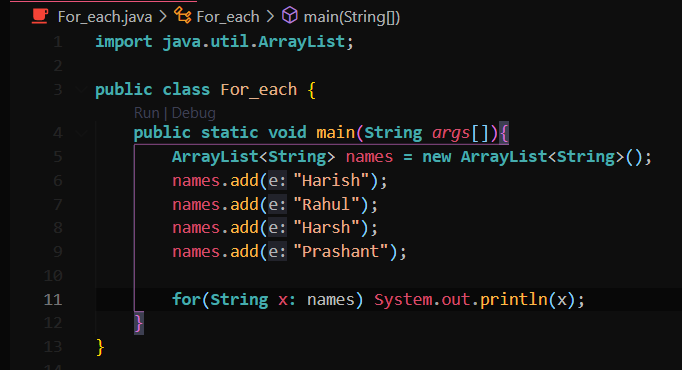


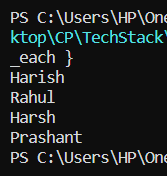


1. 2D ArrayList



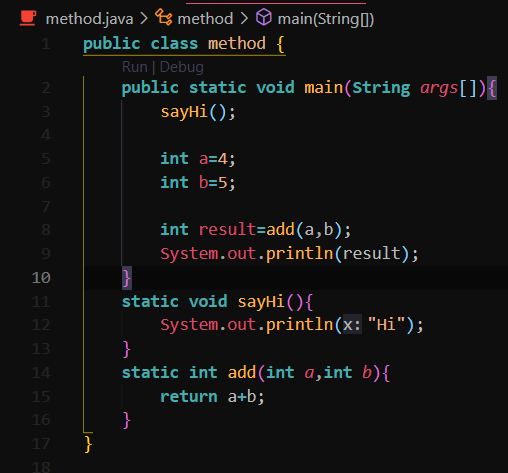
1. For-each Loop

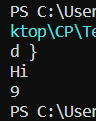




1. Methods

Block of code which can be used again.

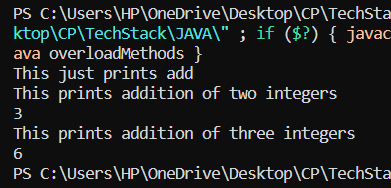




1. Overload methods

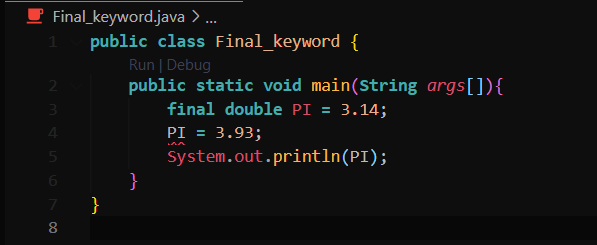
Same method with different parameters

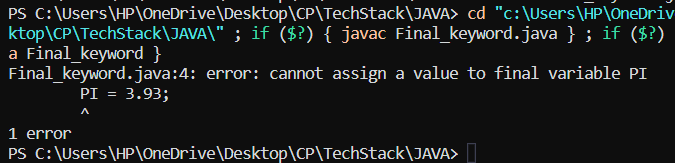




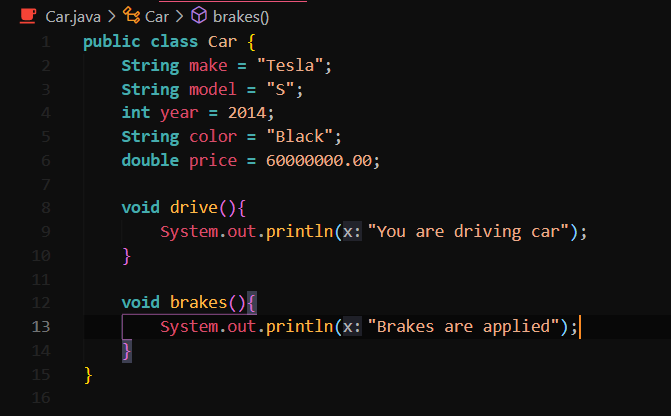
1. Final keyword

To assign a value which will not change

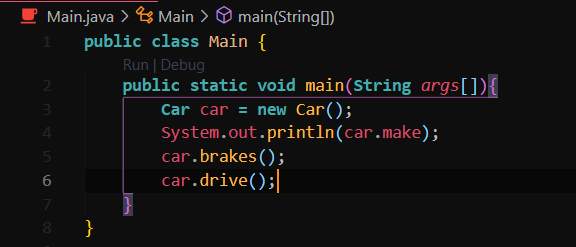




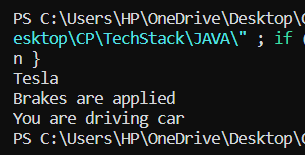
1. Objects OOP



Car.java file

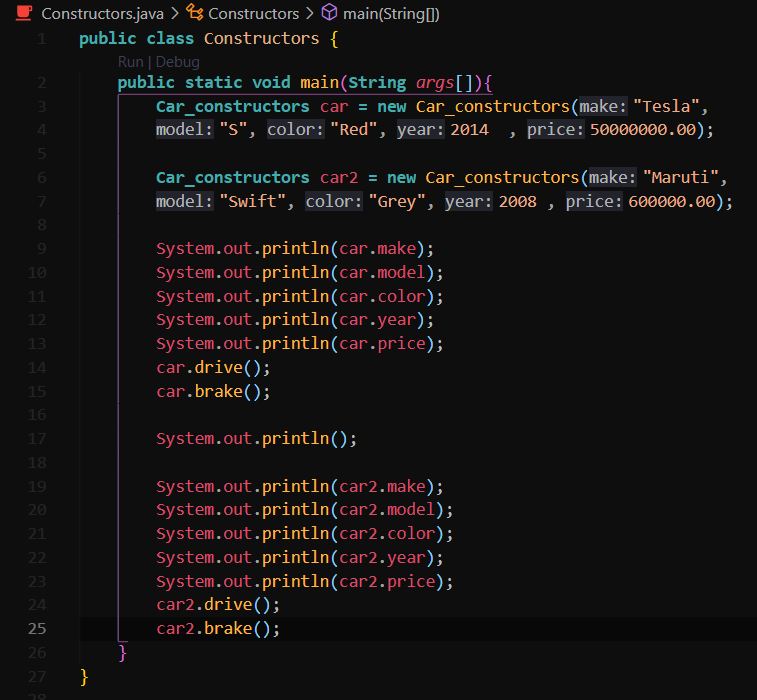


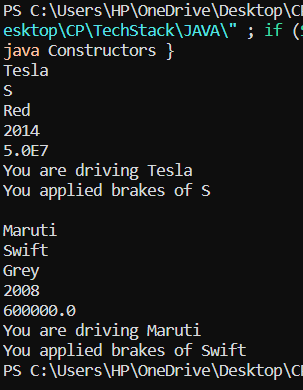
Main.java



1. Constructors

To create objects with different attributes





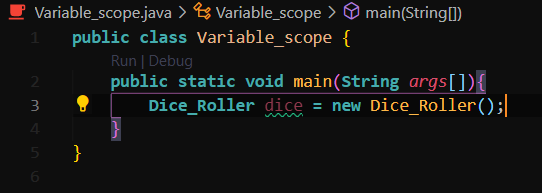
1. Variable Scope

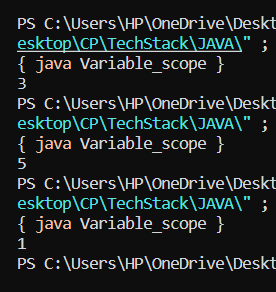
Local = declared inside a method visible only to that method

Global = declared outside a method, but within class visible to all parts of a class

1. Using Local variables

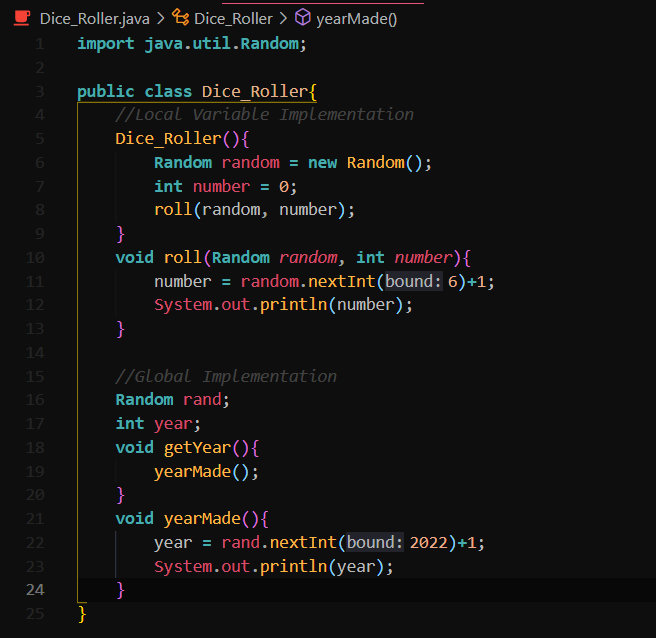


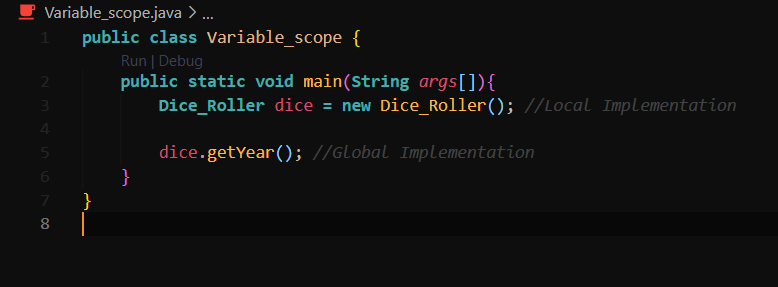




random and number are created locally and then passed as argument to work.

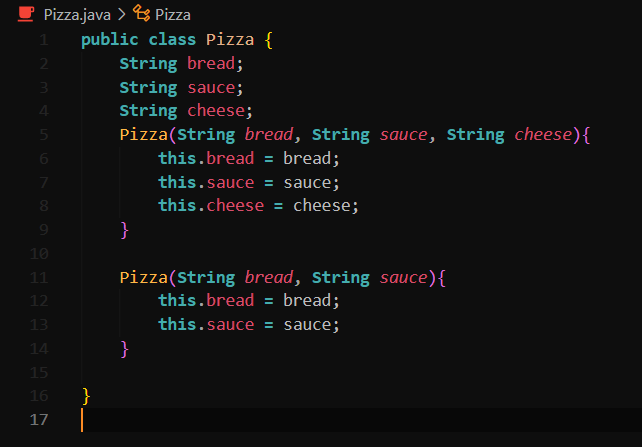
1. Using global

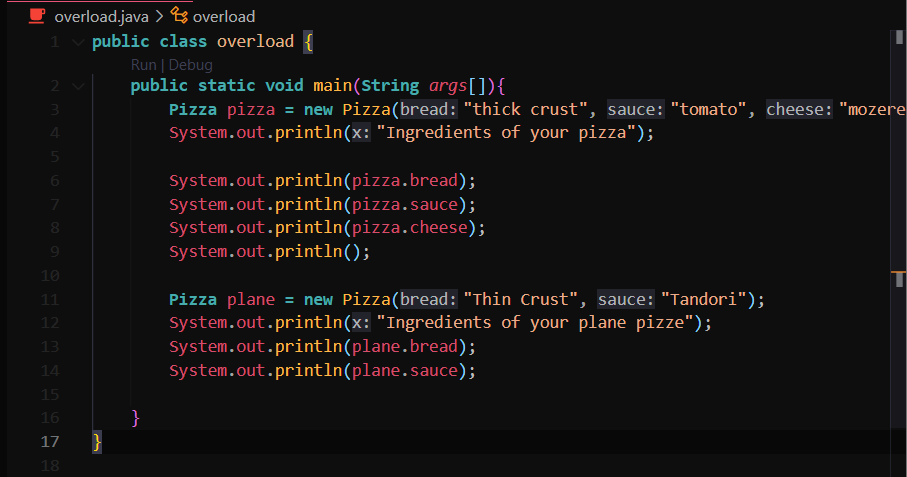




1. Overload Constructors

Multiple constructors within a class with the same name, but have different parameters ( name + parameters = signature)

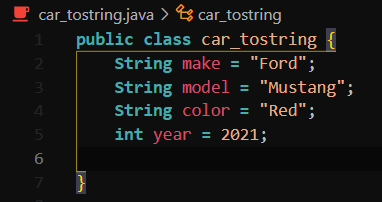


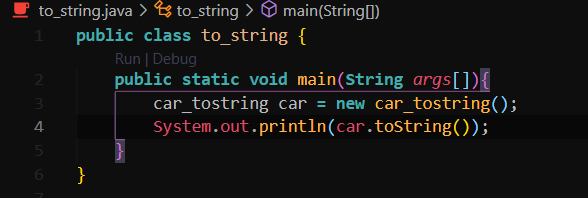




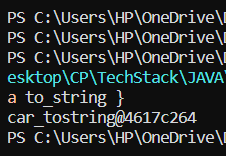
1. toString Method

Special method that all objects inherit, that returns a string that “textually represents” an object. Can be used both implicitly and explicitly.

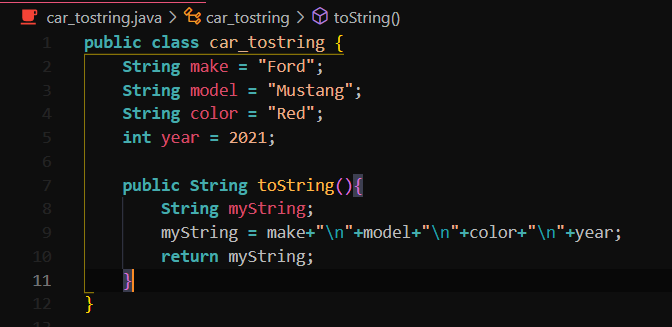


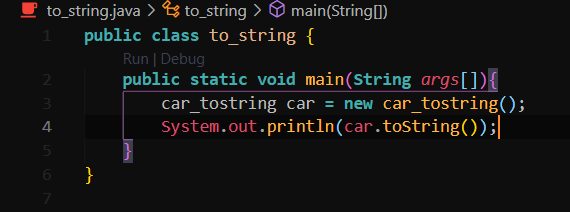


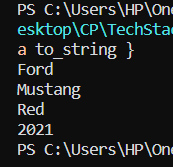
Prints the address of car object



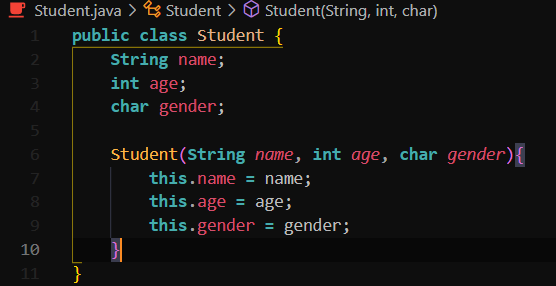
toString() will now be overrided.

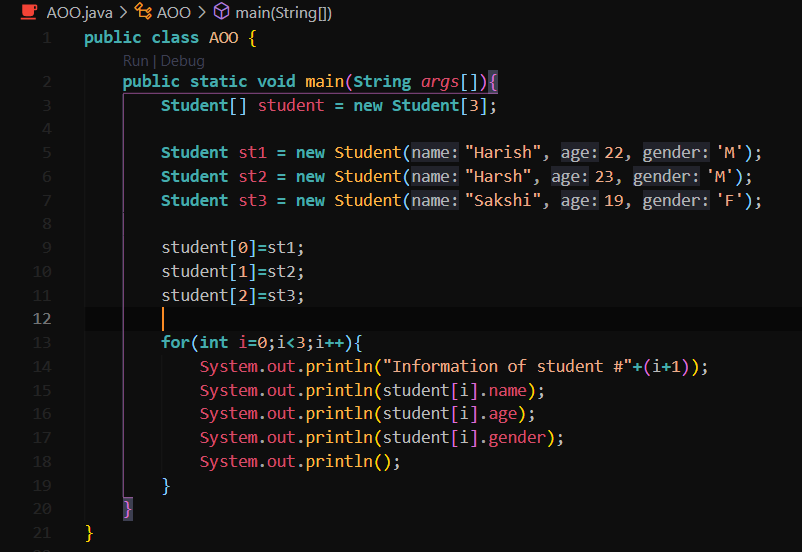


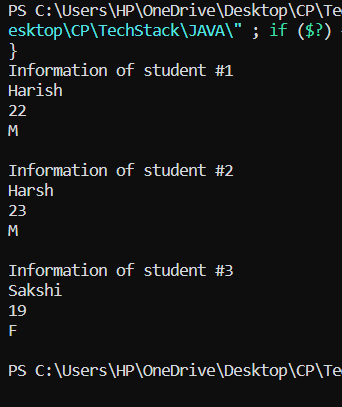




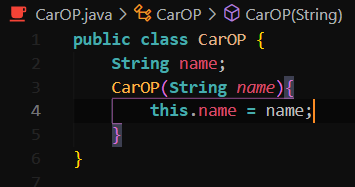
1. Array of objects

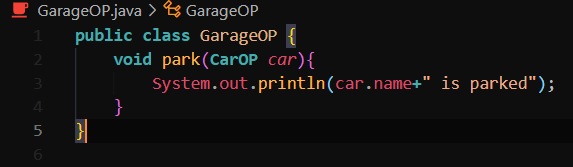




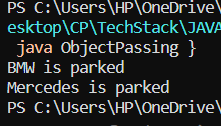


1. Object Passing



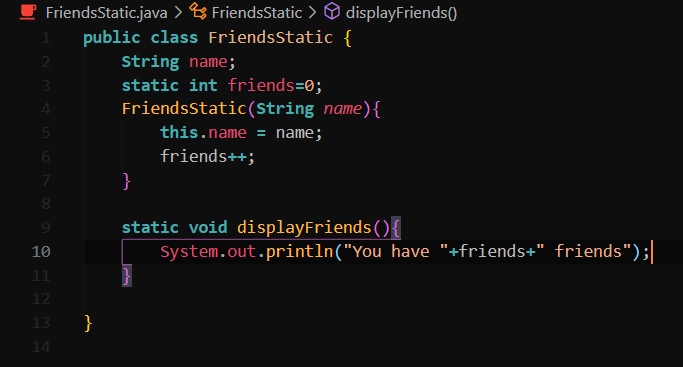


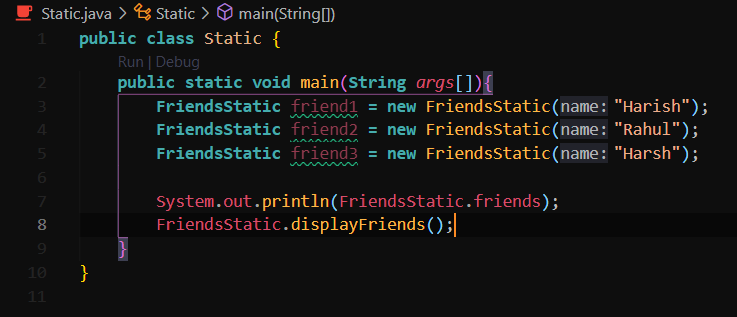


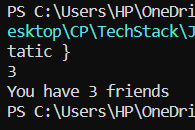


1. Static Keyword

Modifier. A single copy of a variable/method is created and shared. The class owns the static member.



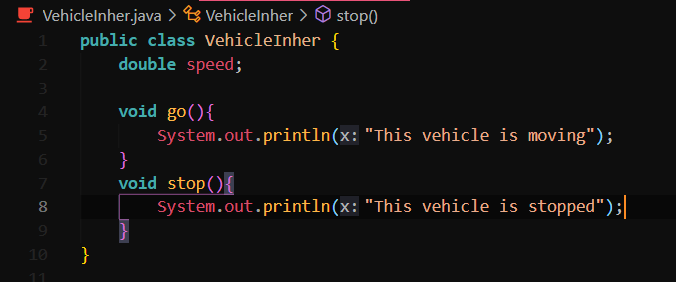


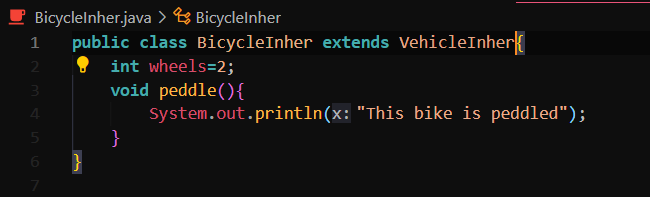
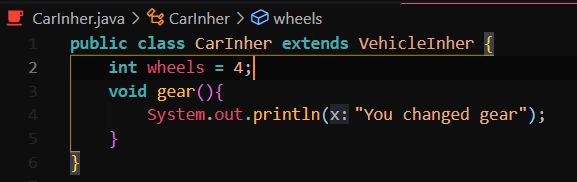


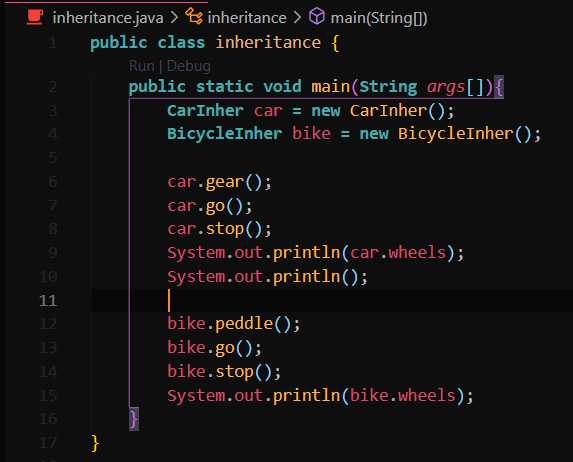
Note: Methods of Math class are also static and are called with Math.method not by creating object of Math class.

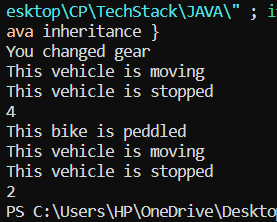
1. Inheritence

Base class can inherit all the members of super class.



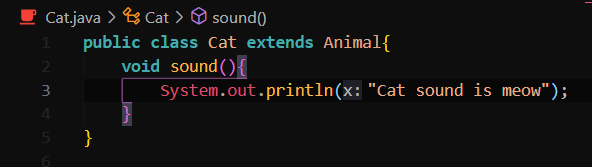
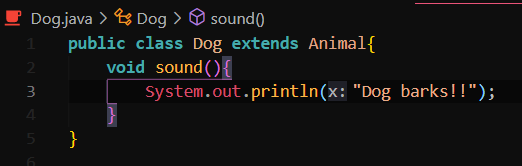
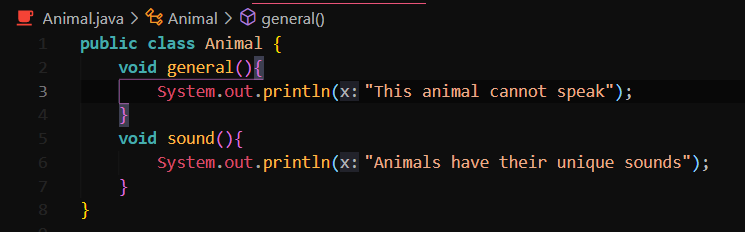


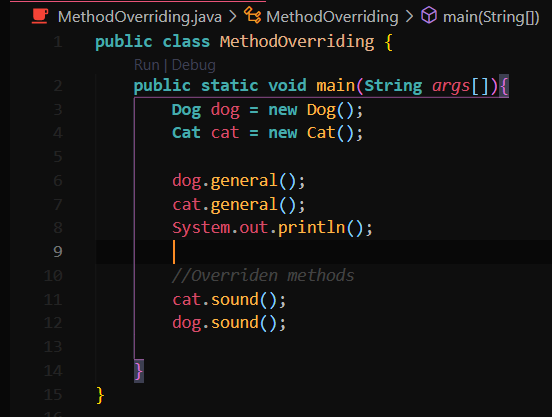


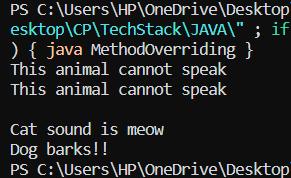


1. Method Overriding

The method of base class overrides the method of super class

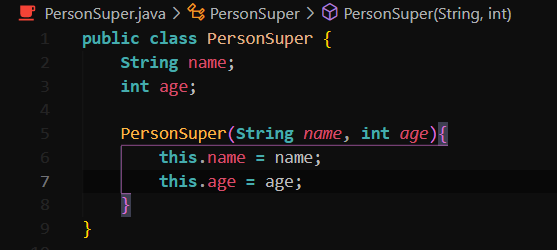


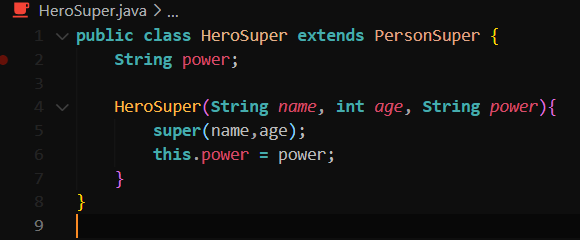


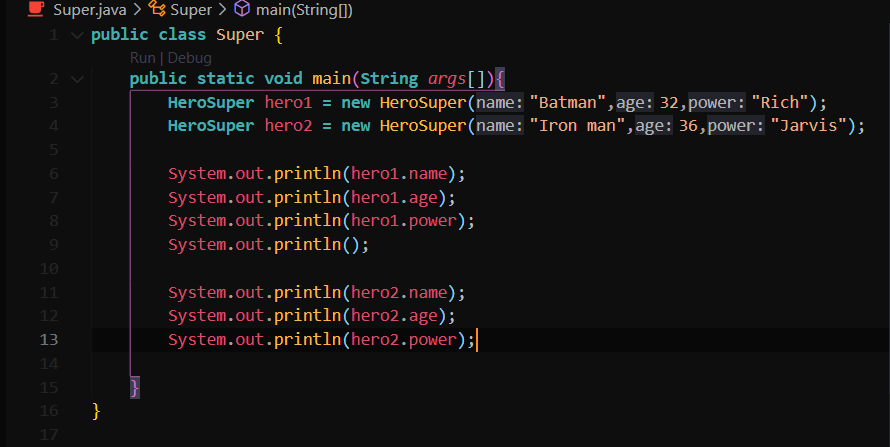


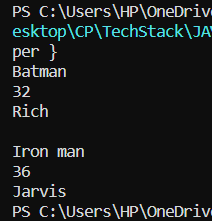
1. Super Keyword

Keyword refers to the superclass (parent) of an object very similar to the this keyword



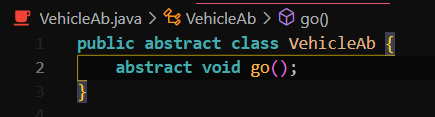


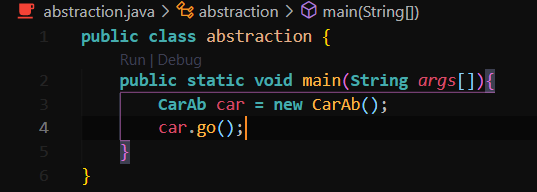
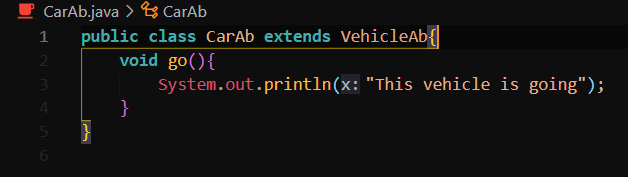


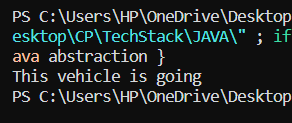


1. Abstraction

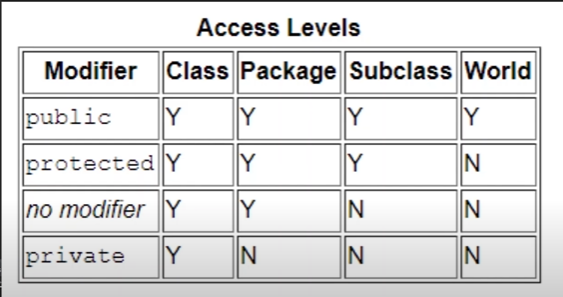
Abstract classes cannot be instantiated but they can have a subclass abstract methods are declared without an implementation





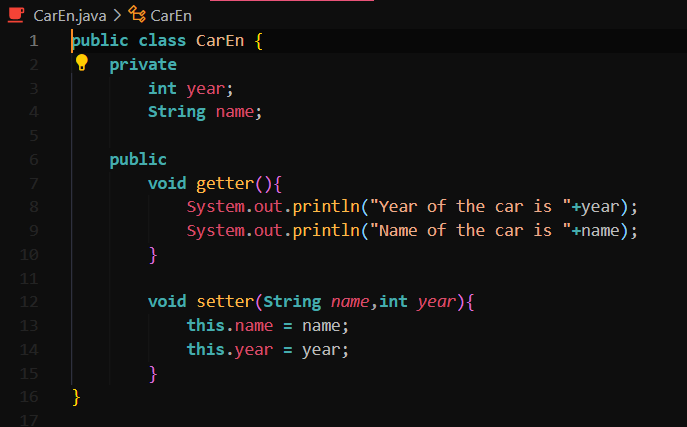


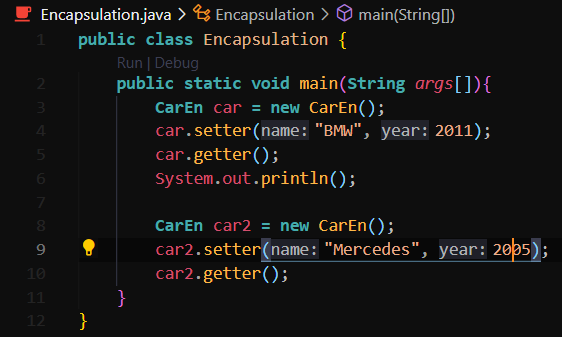
1. Access Modifiers

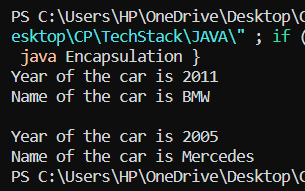


1. Encapsulation

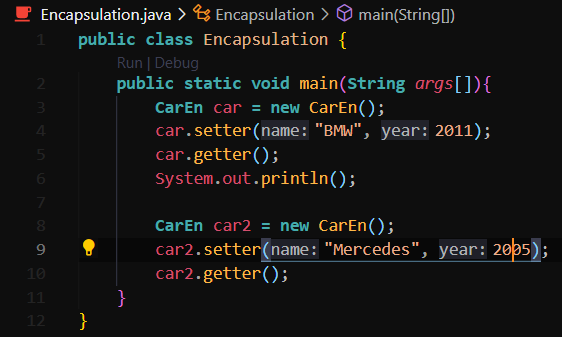
Attributes of a class will be hidden or private, Can be accesed only through methods (getters and setters). We shoud make attributes private if we don’t have a reason to make them public.

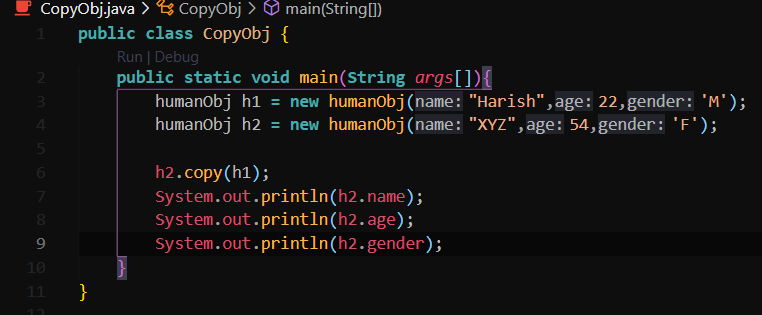


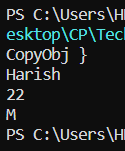




1. Copy Objects

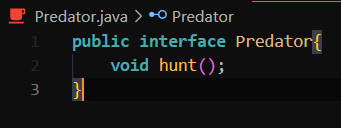


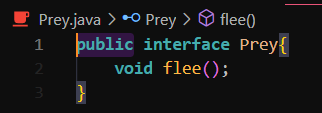


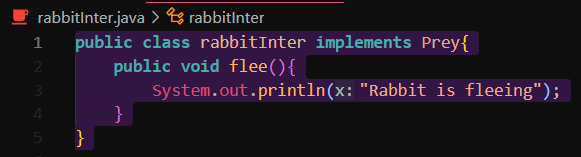


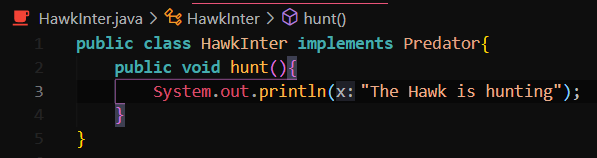
1. Interface

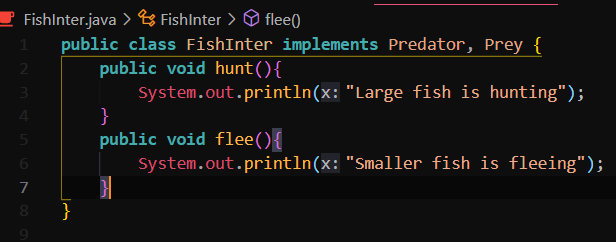
A template that can be applied to a class. Similar to inheritance, but specifies what a class has/must do. Classes can apply more than one interface, inheritance is limited to 1 super class.

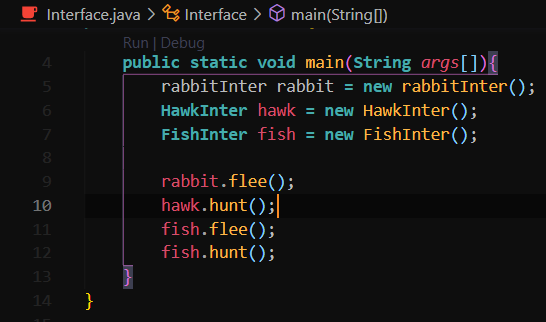


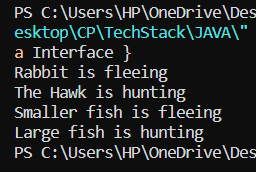








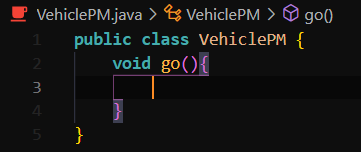


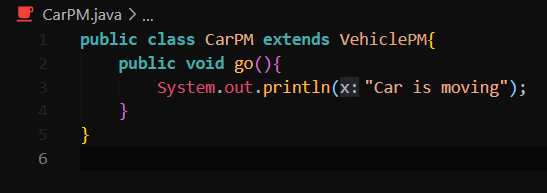


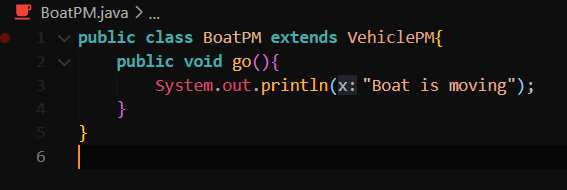
1. Polymorphism

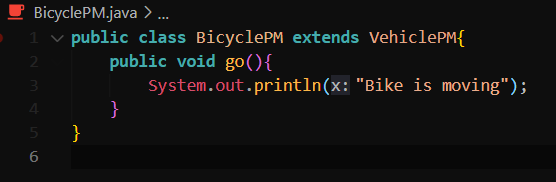
Poly: Many + morph: Form

The ability of an object to identify as more than one type

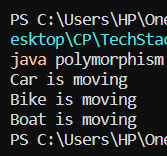






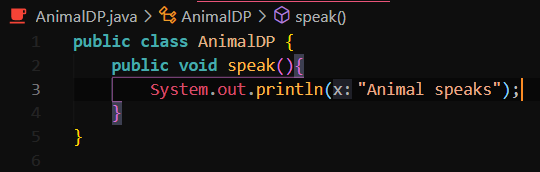


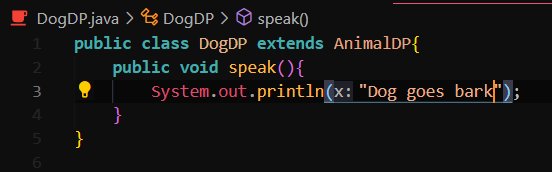


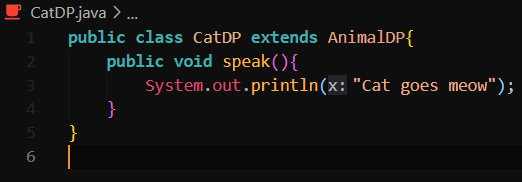


1. Dynamic Polymorphism

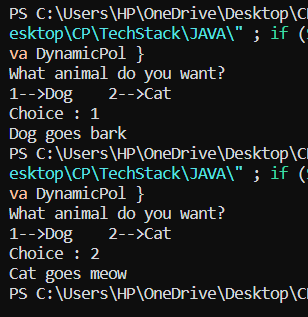
After compilation, it is decided which method to call.





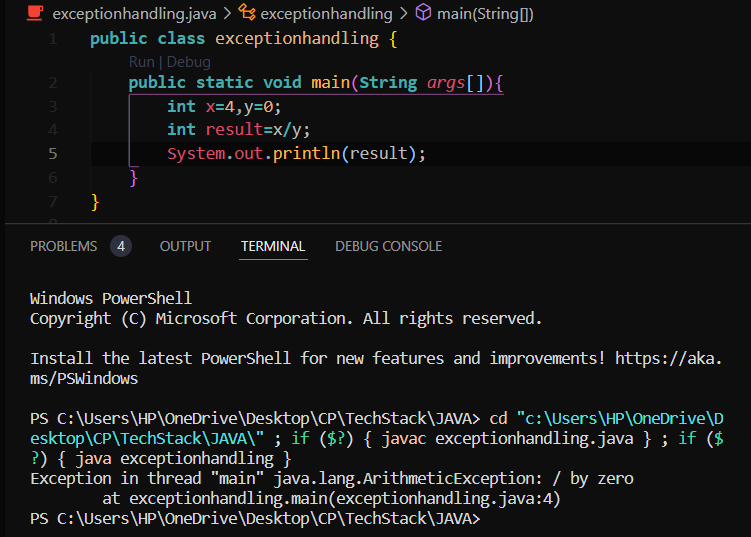






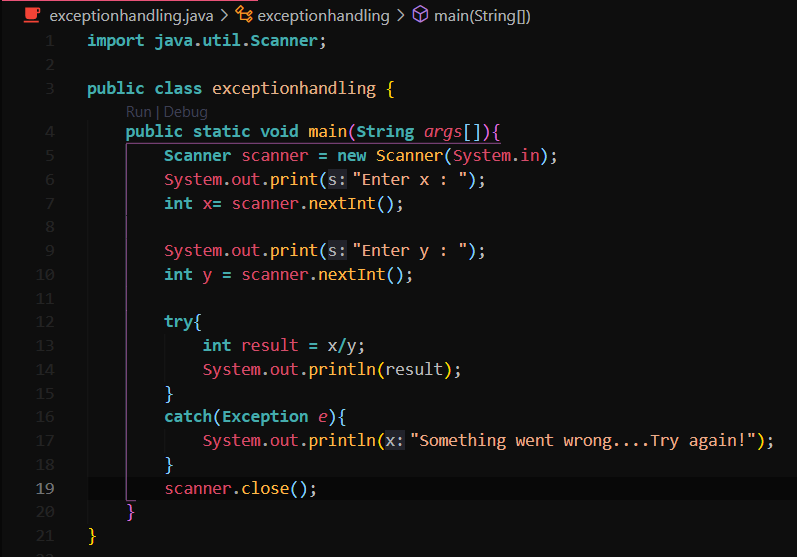
1. Exception Handling

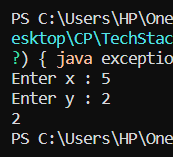
An event that occurs during the execution of a program that, disrupts the normal flow of instructions

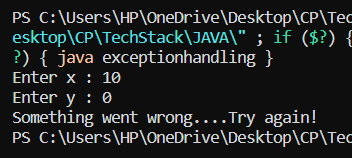


Divide by zero exception

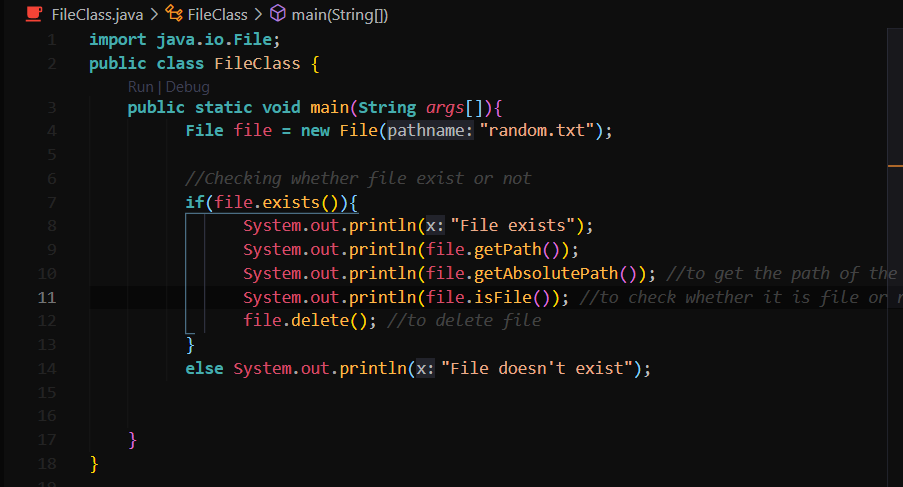
Handling this exception is done using try catch.

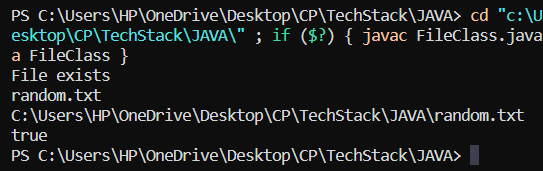




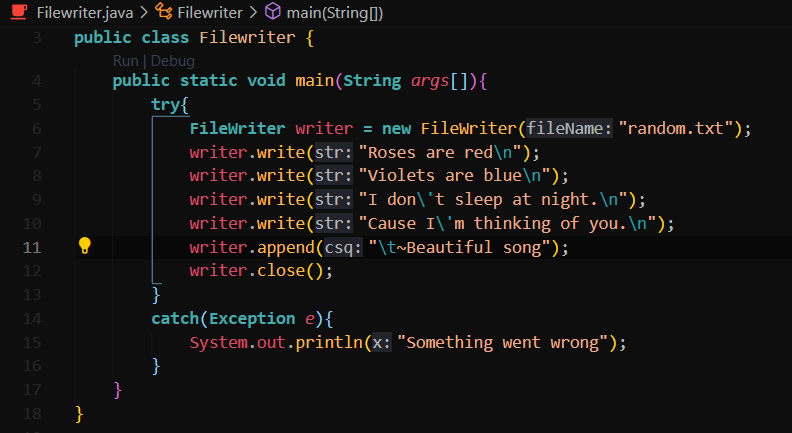


1. File class



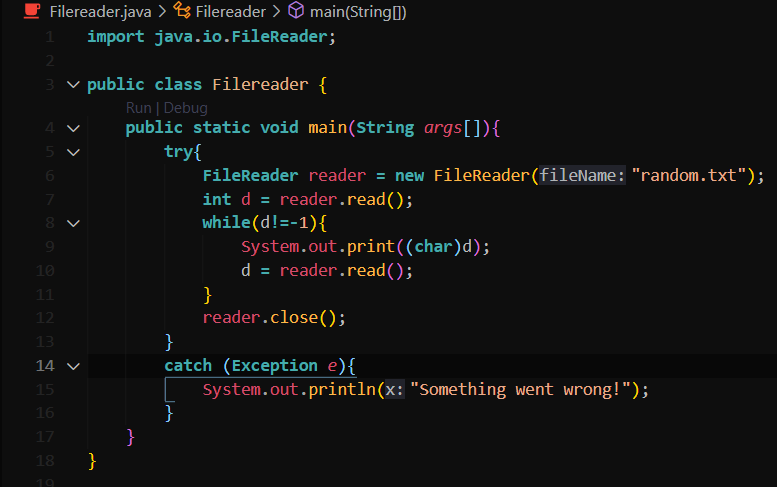


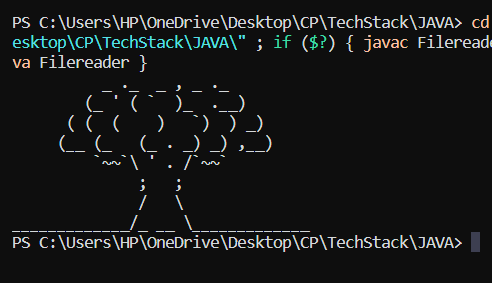
1. FIleWriter



1. FileReader

Read the contents of a file as a stream of characters. One by one read() returns an int value which contains the byte value when read() returns -1, there is no more data to be read



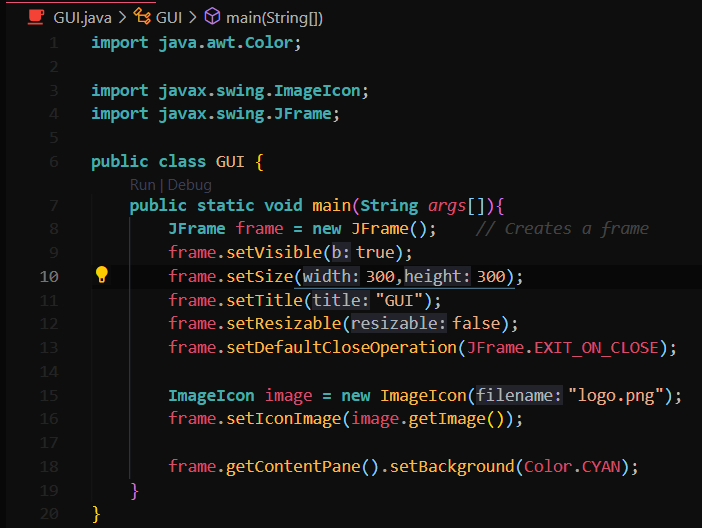


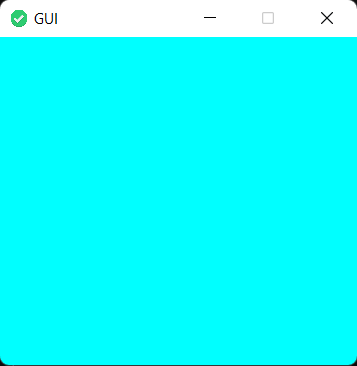
1. Audio

****

1. GUI

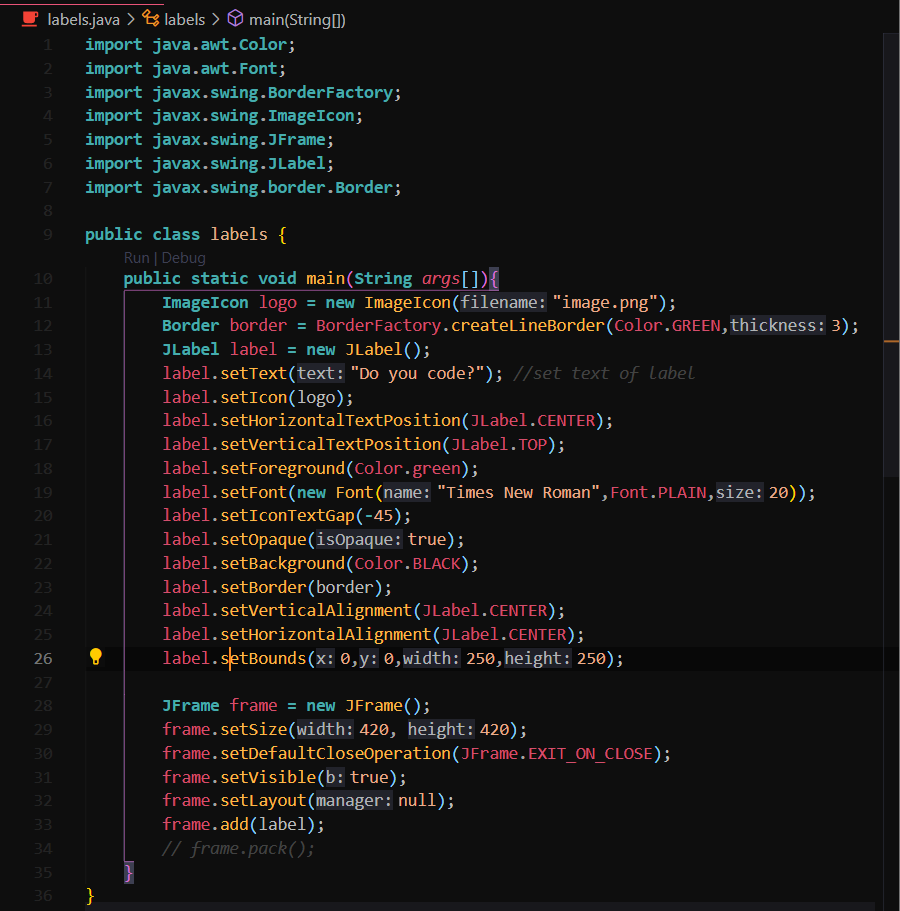
JFram is a GUI window to add components to.

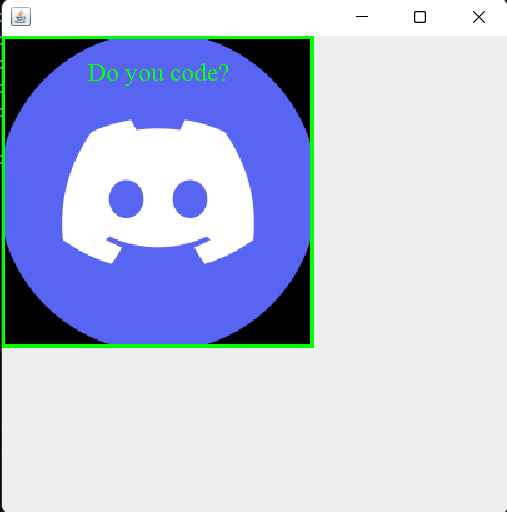




1. Labels

A GUI display area for a string of text, an image, or both





1. Panels

A GUI component that functions as a container to hold other components