Software requirement

1. Java 17 / JDK 17
2. Eclipse IDE

Java: Platform independent & Object Oriented Programming language

Platform independent: It can run java programs on multiple platforms without recompiling

Object Oriented Programming: You create real world entities in the application & these real world entities are called as objects, an object would interact with another object to complete its task

Object will have two things

1. properties: what it has
2. behaviors: what it does

Customer: customer\_id, name, dob, phone, emailId are the properties

deposit(), withdraw(), getDetails() are the behaviours

Employee: employee\_id, name, salary, phone

blockCards(), getCustomerDetails(), registerCustomer()

Building blocks of an object oriented programming language

1. class: It is blueprint/template for an object
2. object: It is an instance of the class

How to write java programs

You create java files & write java classes in it

class Employee {   
 properties : int id; String name; double salary;   
 behaviors : void registerCustomer() { …. } void blockCards() { ….. }  
}

Eclipse: It is an IDE (Integrated Development Environment) which speeds up the development

Workspace: It is a location where all the projects will be saved from eclipse

Java Fundamentals

1. datatypes
2. operators
3. conditional statements
4. loops
5. arrays
6. classes & objects

Datatypes: These are used to specify the type of data a variable stores

|  |  |  |
| --- | --- | --- |
| Datatypes | Size in bytes | default value |
| byte | 1 | 0 |
| short | 2 | 0 |
| int | 4 | 0 |
| long | 8 | 0 |
| float | 4 | 0.0 |
| double | 8 | 0.0 |
| char | 2 | ‘’ (empty character) |
| boolean | 1 | false |

byte, short, int & long stores whole numbers  
float & double stores floating point numbers (real numbers)  
char: to store single character, like M, F, Y, N  
boolean: true or false

How to declare variables:

Syntax: datatype variable = value;

ex:   
int id = 1234;  
double salary = 50020.0  
char gender = ‘M’; // char must use single quote only, double quotes are used in String  
boolean isAdmin = true;

Operators in Java

Arithmetic operators: +, -, /, \*, =

Comparison operators: <=, >=, ==, !=

Logical operators: &&, ||

Increment & Decrement operators: ++, -- (double minus)

Arithmetic operators are useful incase of calculations

double amount = 5000;  
double balance = 15000;  
balance = balance – amount;

double price = 1000;  
double discount = 0.1;  
double total = (price – (price \* 0.1)) \* 1.18; // (1000 – 100) \* 1.18 = 900\*1.18 = 1062

Comparison operators are useful incase of comparisons

int id = 1234; // id gets the data from the UI  
int employeeId = 1234; // employeeId gets the data from DB  
boolean id == employeeId; // id = true

Logical operators are useful to compare multiple expressions

int x = 10;  
int y = 20;  
int z = 30;  
boolean r = (x < y) || (x > z); // 10 < 20 || 10 > 30 // r = true  
boolean s = (x < y) && (x < z); // 10 < 20 && 10 < 30 // s = true

Increment & Decrement operators are useful in case of increasing or decreasing the values

int counter = 0;

counter++; // counter = 1  
counter++; // counter = 2

int anotherCounter = 15;  
anotherCounter--; // anotherCounter = 14

Post and Pre operations in case of increment & decrement

Post increment/decrement: Assigns the value and then increments/decrements

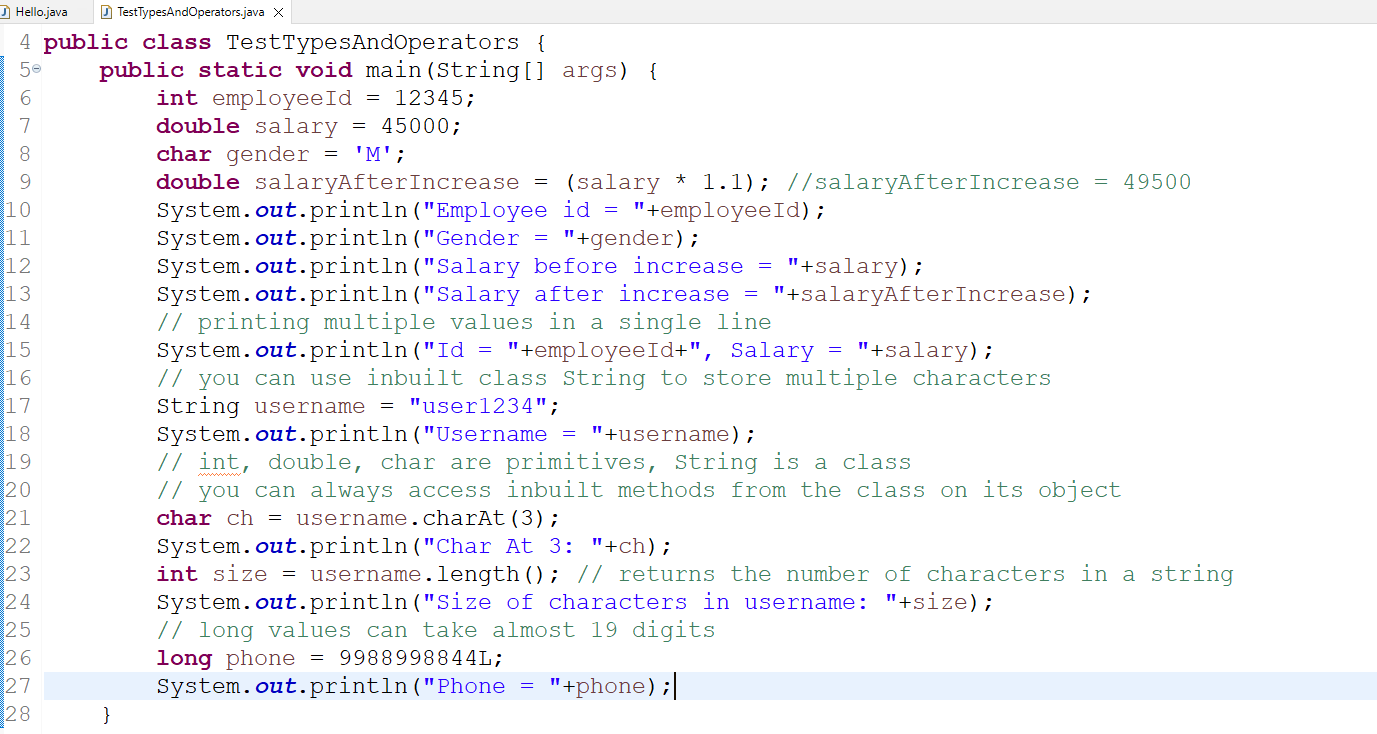
Pre increment/decrement: Increases/Decreases the value and then assigns

int a = 5;  
int b = a++; // b = 5, a = 6

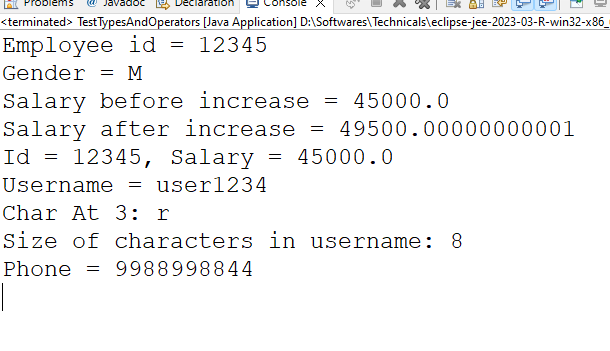
int c = 5;  
int d = ++c; // c = 6, d = 6

counter = 0;  
employeeId = ++counter; // employeeId = 1,

TestTypesAndOperators.java



Output:



Scanner class: It is present in java.util package, which must be imported to use that class, it allows user to give dynamic values from the keyboard / console

Note: Classes from java.lang are auto-imported, classes like String, System, Math need not import

import java.util.Scanner; // if you use eclipse it auto-imports when you use control + space

Java file structure

package statement; // only one package statement in one java file, it must be at the top

import statement; // comes after the package statement, you can zero or more imports

class className { // comes after all the imports

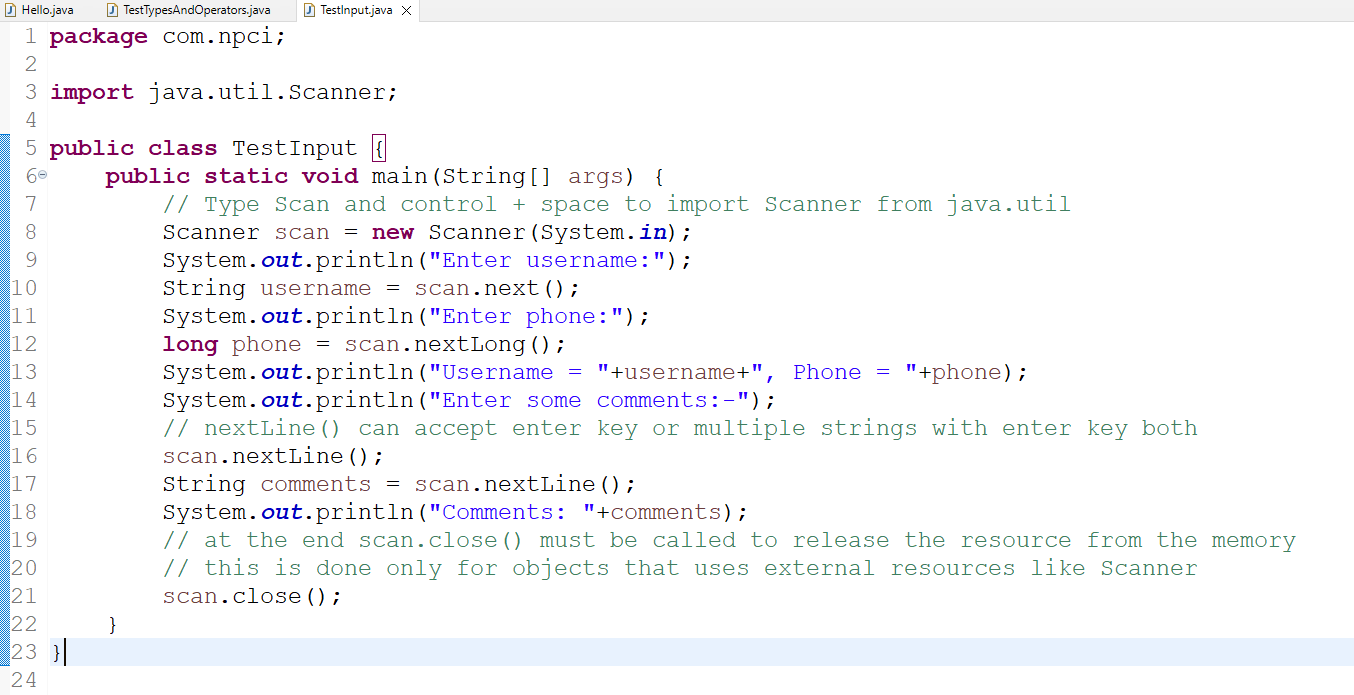
import java.util.Scanner;

We need to create Scanner object to access their methods

Scanner scan = new Scanner( System.in );

scan.nextInt(); // to take int value  
scan.nextDouble(); // to take double value  
scan.next(); // to take a single string  
scan.nextLine(); // to take multiple strings until you press enter key

TestInput.java



Conditional Statements

These are used when you want to run certain block of statements when certain conditions are met.

1. if
2. if else
3. if else if else if … else
4. switch

If:

statement  
if(a > b) {   
 statement  
}

If and Else:

statement  
if(a > b) {   
 statement  
} else {  
 statement  
}  
statement

If, Else If & Else:

statement  
if(a > b) {   
 statement  
} else if (a > c) {   
 statement  
} else if ( a > d) {   
 statement  
} else {   
 statement: executed if all the above conditions are false  
}

switch:

switch(options) {   
 case value1: statement;  
 break;  
 case value2: statement;  
 break;  
 default : statement;  
}

Ex:

marks = scan.nextDouble();  
if(marks >= 70) {   
 // A+  
} else if (marks < 70 && marks >= 60) {   
 // A  
} else if (marks < 60 && marks >= 50) {   
 // B  
} else { // C }

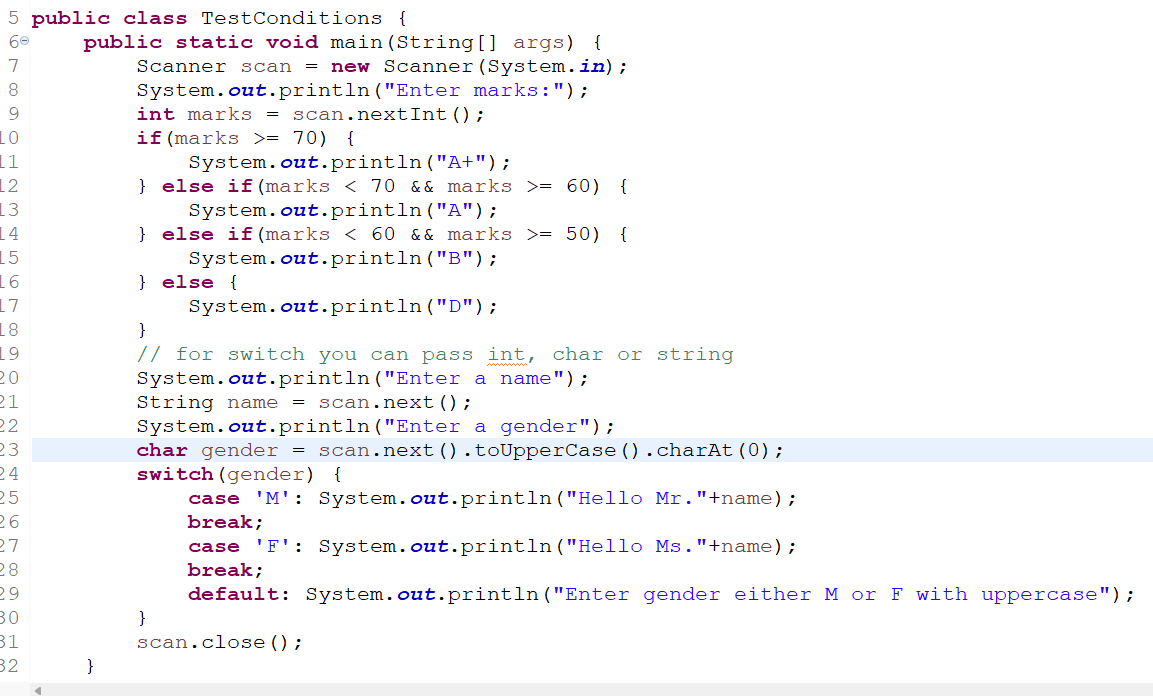
Ex:

option = scan.nextInt();

// switch can compare int, String, char   
switch(option) {  
 case 1: //statements; break;  
 case 2: // statements; break;  
 case 3: // statements; break;  
 default: // statement

}

TestConditions.java



Loops in Java

When you want to run some statements repeatedly until some condition is true you can use loops, we have 3 loops in java

1. for loop
2. while
3. do while

Array: It is a collection to store multiple values  
int[] items = { 3, 1, 4, 5, 2}  
print items[0]; // prints 3  
print items[2]; // prints 4  
print items.length; // prints 5

for loop: When you want to iterate for fixed number of items

for(int index = 0; index < items.length; index++) { // index = 0, 1, 2, 3, 4, at 5 it exits  
 print items[index]; // 3, 1, 4, 5, 2  
}

while loop: When you want to iterate until some condition is true, it first checks the conditions & then executes the loop

do while loop: Similar to while loop, but atleast once the loop is executed & then the condition will be checked

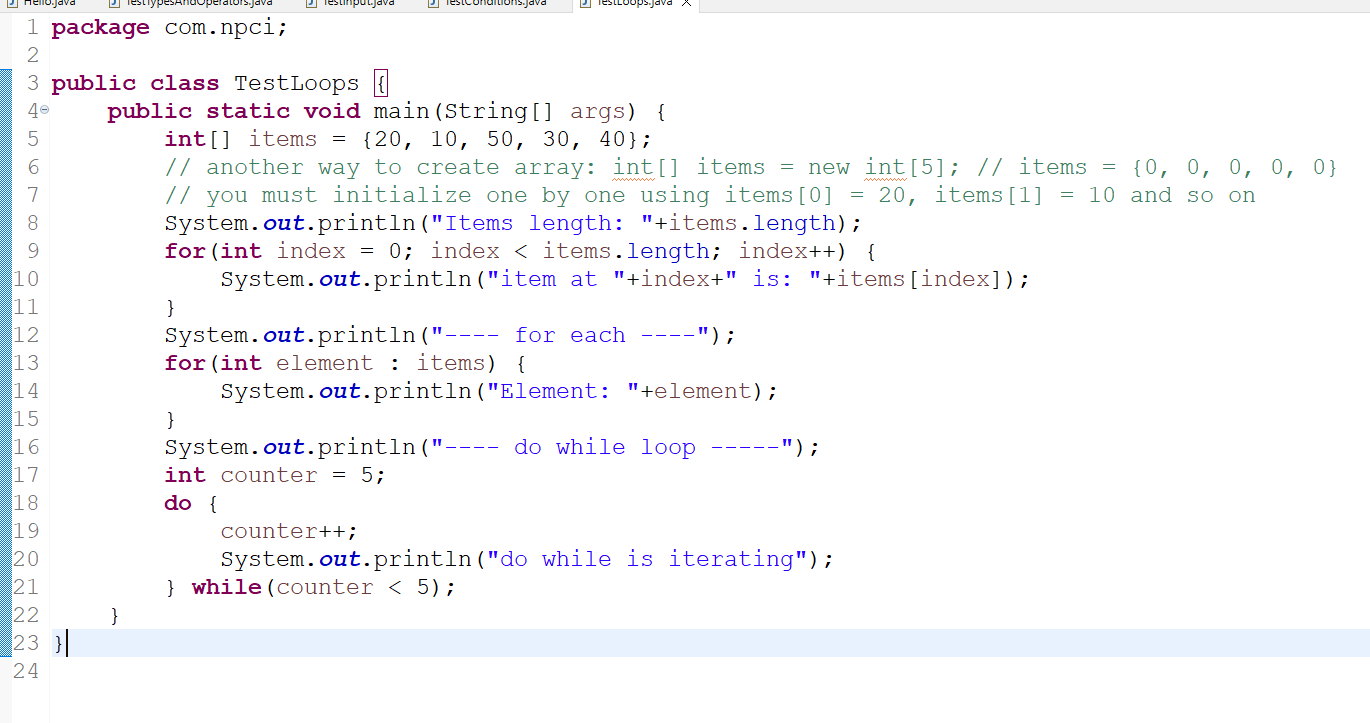
counter = 0;  
while(counter != 5) {   
 counter++;  
}  
counter2 = 5;  
while(counter2 != 5) {  
 counter2++;  
}  
counter3 = 5;  
do {   
 counter3++;  
}while(counter3 < 5);

Note: counter3 becomes 6 because do while will be executed once.

Enhanced for loop / for each: Simplified for loop used on arrays & collections  
Syntax:  
for(type t : collection) {   
 // t is an iterator variable that gets the value from the array  
}

String[] items = {“hello”, “welcome”, “thankyou”}  
for(String s : items) {  
 print s  
}

TestLoops.java



Activity

1. Create an array of some numbers and print their sum of numbers in the array, then maximum and minimum number in the array, use only one loop to perform all the task

ex: if items = {10, -3, 4, 3, 9, 8}, Sum = 31, Max = 10, Min = -3

1. Enter 3 digits number and print each digit in words

ex: Input = 472, Output = Four Seven Two

1. Enter 3 digits number and multiply the highest digit number with the lowest digit number.

ex: Input = 472, Output = 7 \* 2 = 14

Command line argument:

The String[] parameter in the public static void main(String[] args) is the command line argument which can accept the input you pass from the command line before executing the program

These arguments can take input before starting the program, so that the program can use that input and start some servers or connect to some database

java Hello arg1 arg2 arg3 arg4

All these arguments are stored in String[] args (it is optional to pass arguments) but when you use them in some advanced technologies they give lot of benefits

ex: In Spring boot you can pass command line arguments with server port, database credentials as an argument, so that the program can use those command line arguments.

java Hello –server.port=9091 –database.username=root –database.password=12345

String[] args = {“-server.port=9091”, “-database.username=root”, “database.password=12345” }

classes & objects

A class can have following things

1. variables
2. methods
3. constructors