

# Assignment No. 1

Q.1 Write a short note on Java development kit.

→ The Java Development Kit is a software development environment that offers a collection of tools and libraries necessary for developing Java applications. You need the JDK to convert your source code into a format that the Java Runtime environment (JRE) can execute.

The JDK includes the Java Runtime Environment (JRE), an interpreter (Java), a compiler (javac), an archiver (jar), a documentation generator (javadoc), and some other development tools. The Java Runtime Environment itself consists of the Java Virtual Machine (JVM), supporting files, and core classes.

The latest version is JDK 17, which was released in September 2021. Here's a list of its new features:-

- Switch expressions
- Pattern matching for instanceof
- Records
- Helpful NullPointerExceptions
- Text blocks.

Q.2 List & explain the salient features of Java.

→ 1) Simple  
2) Object Oriented

- 3) Portable
- 4) Platform independent
- 5) Secured
- 6) Robust
- 7) Architecture - neutral
- 8) Portable
- 9) Dynamic

4) Pl

ir

or

a

u

5)

1) Simple :- Java is a very easy to learn, and its syntax is simple, clean and easy to understand according to sun, Java language is a simple programming language because:

- Java has removed many complicated & rarely-used features, for example, explicit pointers, operator overloading, etc.

2) Object-oriented :- Java is an object oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects. The basic concept of OOPs are:-

- 1) Object
- 2) Class
- 3) Inheritance
- 4) Polymorphism
- 5) Abstraction
- 6) Encapsulation

3) Portable :- Portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

- 4) Platform Independent :- Java is platform independent because it is different from other languages, like C, C++ etc. which are compiled into platform specific machines while Java is a write once, run anywhere language.
- 5) Secured :- Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:
  - No explicit pointer
  - Java programs run inside a virtual machine sandbox.
- 6) Robust :- Robust simply means strong. Java is robust because:
  - It uses strong memory management.
  - There is a lack of pointers that avoids security problems.
- 7) Architecture-Neutral :- Java is different architecture neutral because there are no implementation dependent features, for example the size of primitive types is fixed.
- 8) Dynamic :- Java is dynamic language. It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages i.e. C, C++ and C#.

Q.3 List and explain the components of Java Virtual Machine.

→ Components of Java Virtual Machine:-

- 1) Class loader:- class loader is used to loads the class files into memory. whenever we executed the java program , class loader loads it first. There are three built in class loader are available in Java:-
  - 1) Bootstrap class loader
  - 2) Extension class loader
  - 3) Application class loader
- 2) Method Area:- Method area stores the all the class level data such as runtime constant pool, fields, method data and code for methods. There is only one method area per JVM.
- 3) Heap:- Heap Memory created when JVM start up. It stores all the objects created during program execution and their corresponding instant variable. It is the run-time data area from which memory for all class instances and array is allocated. There is only one heap area per JVM.
- 4) Stack : Java stack memory stores frames, local variable, method callers , partial results.

Whenever a new thread is created in the JVM, a separate JVM is also created at the same time for that thread.

5) Program Counter (PC) Registers:- Program Counter Register stores address of the currently executing JVM Instruction. JVM supports Multiple threads, so each thread has its own PC register.

6) Native Libraries:- Java Native libraries provides a collection of libraries which are written in other programming languages needed by execution engine.

Q4 Write in detail about different types of operations in Java, category wise quoting their functionality, operands and return type. Give one example statement for each.

→ Different types of operators:-

1) Arithmetic Operator:- Arithmetic operators in Java perform simple and basic mathematical operations, such as subtraction, addition, division, multiplication, on primitive data types. For example,  $m + n$ . Here  $+$  is the arithmetic operator used to add variables  $m$  and  $n$ .

Operator	Description
+	Addition :- It is used to add two variables or values of the operands.
-	Subtraction :- This operator is used to perform subtraction.
*	Multiplication :- It is used to multiply given values.
/	Division :- This arithmetic operator is used to divide the left-hand value with the right-hand values.
%	Modulo Operations :- Also known as Remainder operator, it returns the remainder after division.

Example:-

```
class ArithmeticOperator
{
    public static void main(String arg[])
    {
        double a=20, b=4;
        System.out.println("Addition :" + (a+b));
        System.out.println("Subtraction :" + (a-b));
        System.out.println("Multiplication :" + (a*b));
        System.out.println("Division :" + (a/b));
        System.out.println("Modulus :" + (a%b));
    }
}
```

Output:-

Addition : 24.0

Subtraction : 16.0

Multiplication : 80.0

Division : 5.0

Modulus : 0.0

[2] Assignment Operators : Assignment operator is one of the most common operators, you are using in java program. It is used to assign value on its right to the operand on its left. symbol : '='

operator	description
$+=$	Addition Assignment: It work by adding the current value of the variable on left to the value on the right and then assign result to the operand on the left.
$-=$	Subtraction Assignment: It work by subtracting the current value of the variable on left to the value on the right and then assign result to the operand on the left.
$/=$	Division Assignment: It work by dividing the current value of the variable on left to the value on the right and then assign result to the operand on the left.
$*=$	Multiplication Assignment : It work by multiplying the current value of the variable on left to the value on the right & then assign result on the left.
$\%=$	Modulus Assignment : The current value of the variable on left to the value on the right & then assign remainder to the operand on the left.

Example:

```
class AssignmentOperator
{
    public static void main (String args[])
    {
        int num = 60;
        double a=20, b=30;
        a+=10;
        b-=5;
        System.out.println ("Assignment operator: " + num);
        System.out.println ("Addition Assignment: " + a);
        System.out.println ("Subtraction Assignment: " + b);
    }
}
```

Output:

Assignment operator : 60

Addition Assignment : 30.0

Subtraction Assignment : 25.0

### [3] Increment and Decrement Operators:

Increment operator is used to increase value of operand by one. Decrement operator is used to decrease value of operand by one.

Both operators required only one operand. These operators can be used in either prefix and postfix form.

Operator	Description
++	Increment operator
--	Decrement operator

Program:

```
class Operator
{
```

```
public static void main (String args[])
{
```

```
    int num1 = 20, num2 = 10;
```

```
    System.out.println (num1++);
```

```
    System.out.println (++num1);
```

```
    System.out.println (num2--);
```

```
    System.out.println (---num2);
```

```
}
```

```
y
```

Output :-

20

22

10

18

[4] Relational Operator:- Relational operators are used to determine relationship between two variables, like value of one operand is greater than, equal to or less than another operand.

Operators	Description
==	equal to
!=	not equal to
>	greater than
<	less than
>=	Greater than and equal to
<=	less than and equal to

example:-

```
class Relationaloperator
```

{

```
    public static void main (String args[])
    {
        int a = 10, b = 30, c = 20;
        System.out.println ("a < b: " + (a < b));
        System.out.println ("a > b: " + (a > b));
        System.out.println ("a == b: " + (a == b));
        System.out.println ("a != b: " + (a != b));
    }
}
```

output :-

a < b : true

a > b : false

a == b : false

a != b : true

## 5) Logical operators:-

Logical operators are used to perform logic AND, OR and NOT operation. Logical AND return true result if both the expressions are true. Logical OR return true result if either one expression is true. Logical NOT return true if expression is false and vice versa

operator	Description
&&	Logical AND
	Logical OR
!	Logical NOT

example:-

```
class Logicaloperator
```

{

```

public static void main (String arg [ ] ) {
    int a = 10, b = 30, c = 20;
    System.out.println ("Logical AND: " + (a < c && a < b));
    System.out.println ("Logical OR: " + (a > c || a < b));
    System.out.println ("Logical OR: " + (a > c & a < b));
    System.out.println ("Logical NOT: " + ! (a > b));
}

```

Output :

Logical AND: true

Logical OR: true

Logical OR: false

Logical NOT: true

### (6) Bitwise Operator:-

Bitwise operator operate on binary value. It returns result by performing manipulation on individual bit of a number. In Bitwise OR operation, it performs bit by bit OR operation of input value and it return 1 if either of the bits is 1 otherwise it return 0.

### (7) Conditional Operator:-

Ternary operator is similar to if-else then-else statement in java. A ternary operator evaluates the test condition and execute the block of code based on result of the condition.

Syntax :- condition ? expression 1 : expression 2

Q.5

What are the primitive data types in java? Briefly explain their size, range and other details.

→ When we define a variable in java, we must need to define what kind of value that variable will stored. This will be defined with the help of data types.

Java programming language defines eight primitive data types: byte, short, int, float, double, char, boolean. A primitive type is predefined in java language. primitive size of these primitive data type is fixed, it does change from one operating system to another. This type can be placed into four group.

Integers: This group includes byte, short, int, and long, which represents whole numbers. All these integers support signed, positive and negative values.

Floating-Point numbers :- This group include float and double data type, which represent numbers with fractional value.

Character:- This group include char data type.

Boolean:- This group include Boolean data type, which includes binary value like true, false.

Data Type	size	Default value
byte	1 byte	0
short	2 byte	0
int	4 byte	0
long	8 byte	0
float	4 byte	0.0F
double	8 byte	0.0D
char	2 byte	'\u0000
boolean	1 bit	false

1) byte :- It is a smallest integer type. The byte data type is 8 bits (1 byte) signed two's complement integer. It has range lies between -128 to 127. Its default value is 0.  
 syntax of declaring byte variable:  
 byte variable1;

2) short : It is signed 16 bits (2 bytes) two's complement integer. It has value range lies between -32768 to 32767. Its default value is zero.  
 syntax;  
 short t,s;

3) int : The int data type is 32 bits (4 bytes) signed two's complement integer. It has the value range lies between  $(2^{31})$  to  $(2^{31}-1)$ . Its default value is zero. It is most commonly used data type.

syntax:

int a,b;

int c = 2;

4) long :- The long data type is 64 bits (8 byte) signed two's complement integer. It has value range lies between  $-(2^{63})$  to  $(2^{63}-1)$ . Its default value is zero.

5) float :- The float data type is single precision 32 bits (4 byte) floating point. It is useful storing smaller precision but not suited for large degree of precision. Its default value is 0.0f.

Syntax :-

float f = 3.14f, a;

6) double :- The double data type is double precision 64 bits (8 bytes) floating point. It is useful for storing floating point values with large precision. Its default value is 0.0d.

Syntax :

double d = 45.78521457

7) char :- The char is 16 bits (2 byte) Unicode character. It has the value range lie betn 0 to 65535. This data type is used to stored characters.

Unicode defines a fully international character set that can represent all of the characters that are available in all human languages. Its default value is '\u0000'.

Syntax :

char c = 'g'.

Q.1 Boolean :- The Boolean data type is used to store logical value. It stores only two possible values: true and false. It represents only one bit of information. Values of Boolean variable can not be converted into any other type. Its default value is false.

Q.2 Explain about memory management in java with reference to stack and heap.

→ Memory management in java refers to the process of allocating and freeing up space for objects. Java automatically manages memory. The "garbage collector" is an autonomous memory management technique used in Java.

1) Heap memory is used by all the parts of the application whereas stack memory is used only by one thread of execution.

2) whenever an object is created, it's always in the heap space and stack memory contains the reference to it.

3) Stack memory only contains local primitive variable to objects in heap space.

4) objects stored in the heap are globally accessible whereas stack memory can't be accessed by other threads.

5) Memory management in stack is done in LIFO manner whereas it's more in LIFO manner whereas it's more complex in heap memory because it's used globally. Heap memory is divided into young Generation, old Generation etc. more.

- i) Stack memory is short-lived whereas heap memory lives from the start to the end of application execution.
- ii) Stack memory size is very less when compared to Heap Memory. Because of simplicity in memory allocation (LIFO). Stack memory is very fast when compared to heap memory.

Q.7 Explain the terms: narrowing, widening.

→ Narrowing :- Narrowing, also known as downcasting / casting, a conversion that is explicitly performed in the following situations-

- Narrowing a wider / bigger primitive type value to a smaller primitive type value.
  - Narrowing a superclass reference to a subclass reference, during inheritance.
- widening

Narrowing is explicitly performed when a wider / bigger primitive type value is assigned to smaller primitive type value. This is also known as downcasting / casting a bigger primitive value to a small primitive value.

Widening :- Widening, also known as upcasting, is a conversion that implicitly takes place in the following situation :-

- Widening takes place when a smaller primitive

type value is automatically accommodated in a larger (wider) primitive data type.

• Widening also takes place when a reference variable also takes place when a reference variable of a sub-class is automatically accommodated in a reference variable of its superclass.

Q.8 Write a in detail about static keyword.

→ The static keyword in Java is mainly used for memory management. The static keyword in Java is used to share the same variable or method of a given class. The users can apply static keywords with variables, methods, blocks and nested classes. The static keyword belongs to the class than an instance of the class. The static keyword is used for a constant variable or a method that is the same for every instance of a class.

The static keyword is a non-access modifier in Java that is applicable for the following:

1. Blocks
2. Variables
3. Methods
4. Classes

Characteristics of static keyword:

1) Shared memory Allocation:- static variables and methods are allocated memory space

only once during the execution of the program.

- 2) Accessible without object instantiation: static members can be accessed without the need to create an instance of the class. This makes them useful for providing utility function and constants that can be used across the entire program.
- 3) Associated with class, not objects: - static numbers are associated with the class; not with individual objects.

Q.9 Write a short note on access specifiers in java.

→ The access specifiers / modifiers in java specifies the accessibility or scope of a field, method, constructor, or class. We can change the access level of fields, constructors, methods, and class by applying applying the access specifier modifier on it.

There are four types of java access specifiers: -

1] Private: The access level of a private modifier is only within the class. It can not be accessed from outside the class.

2] Default: - The access level of a protected modifier is only within the package. It cannot accessed from outside the package. If you do not specify any access level, It will be the default.

3) Protected : The access level of a protected modifier is within the package be accessed from outside the package.

4) Public :- The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.