

## **UNIT-I**

### **Introduction to Programming**

Program is defined as set of instructions. We programmers give instructions to system to fulfil the requirement of users, is called programming. We use computer languages to give instructions to computer (for programming). The computer languages are classified into two types.

Low level languages

High level languages

The languages that are capable of accessing processor features directly are called low level languages. w.r.t. systems, it is said that system understands low level languages. But low level languages have a problem that a program designed in LLL for a specific modal processor cannot be executed on other modal processors because of change in features. This is called hardware dependency. To overcome this problem, high level languages are developed. But HLL cannot access processor features directly, (w.r.t. system it is said system doesn't understand HLL). So they cannot be executed directly. For execution the code written in HLL must be converted into LL code/ machine code using compilers or interpreters. Compilers, interpreters are software mechanisms used to convert HL code into PROCESSOR APPROPRIATE LL code.

The compilers convert HL code into LL code at once and stores as a file on hard disk. The interpreters convert HL code into LL code line after line and pass into processor for execution. They won't create a file. When compared compilers are faster than interpreters. The languages C, C++, Java etc are provided with two phases of conversion of HL code into LL code to provide SECURITY TO HL CODE. The C has two compilers and the java has one compiler and one interpreter.

Runtime Environment:

Every SW has its own environment, and the environment of programming languages is called RUNTIME ENVIRONMENT. (In short called RE). The RE of languages include no. of components using which we give instructions to system and under control of this RE the programs executes. The RE components are also called TOKENS of that language.

-----  
\*\* this is introduction for next content.  
-----

### **OOPs Concepts and Java Programming**

Introduction to Object-Oriented Programming:

All high level languages support user defined functions. The user defined functions are used to increase modularity in the program. When we develop a project, it might include thousands of lines of code. As it becomes difficult of typing entire program in one function, all high level languages support dividing programs into functions.

General scope of variables in programming while using functions is such that, we can declare variables either globally or locally. The variables declared with in a function definition are

called local variables to that function and they are accessible only within that function. The variables declared outside of all functions are called global variables and they are accessible to all the functions from the location of their declaration. But as global variables are accessible in all the functions, there is a chance of misuse of them, which leads to data corruption. So it is suggested to programmers to reduce usage of global variables. In this case, we will be using only local variables, but local variables are not accessible in other functions. They have their SCOPE only within that function in which they are declared. So functions provide a facility of passing values from one function into another through ( ) as arguments from function call into function definition. But as there is a facility of passing arguments there is a chance of passing wrong argument, which leads to data corruption.

To provide SECURITY to data, a new technique of programming is designed called OOPS (Object Oriented Programming System). In this programming system, the variables and functions are clubbed into one entity called object and when a function is called for an object, the function automatically processes the variables with which it is clubbed with. And there is no requirement of passing as arguments.

In this scenario, as there is no requirement of passing arguments, there is no chance of passing wrong argument. Thus the OOPS provide data security. In structured programming and previous generation of languages, the concentration was only on functions. But in OOPS, the concentration is more focused on data and also on methods. Moreover, the objects in the program are treated as practical live objects. We can understand and manipulate them as a real world object. We can treat and interact with them as real.

-----  
Q. Give an introduction to OOPS Programming.  
-----

### Procedural and Object Oriented Programming Paradigm :

All high level languages support user defined functions. The user defined functions are used to increase modularity in the program. When we develop a project, it might include thousands of lines of code. As it becomes difficult of typing entire program in one function, all high level languages support dividing programs into functions.

General scope of variables in programming while using functions is such that, we can declare variables either globally or locally. The variables declared with in a function definition are called local variables to that function and they are accessible only within that function. The variables declared outside of all functions are called global variables and they are accessible to all the functions from the location of their declaration.

But as global variables are accessible in all the functions, there is a chance of misuse of them, which leads to data corruption. So it is suggested to programmers to reduce usage of global variables. In this case, we will be using only local variables, but local variables are not accessible in other functions. They have their SCOPE only within that function in which they are declared. So functions provide a facility of passing values from one function into another through ( ) as arguments from function call into function definition. But as there is a facility of passing arguments there is a chance of passing wrong argument, which leads to data corruption.

To provide SECURITY to data, a new technique of programming is designed called OOPS (Object Oriented Programming System). In this programming system, the variables and functions are clubbed into one entity called object and when a function is called for an object, the function automatically processes the variables with which it is clubbed with. And there is no requirement of passing as arguments.

In this scenario, as there is no requirement of passing arguments, there is no chance of passing wrong argument. Thus the OOPS provide data security. In structured programming and previous generation of languages, the concentration was only on functions. But in OOPS, the concentration is more focused on data and also on methods.

Moreover, the objects in the program are treated as practical live objects. We can understand and manipulate them as a real world object. We can treat and interact with them as real world entity. So this approach of programming will have more simple, practical.

---

Q. Explain Procedure oriented paradigm and object oriented paradigm.

---

Procedural Programming can be defined as a programming model which is derived from structured programming, based upon the concept of calling procedure. Procedures, also known as routines, subroutines or functions, simply consist of a series of computational steps to be carried out. During a program's execution, any given procedure might be called at any point, including by other procedures or itself.

Object oriented programming can be defined as a programming model which is based upon the concept of objects. Objects contain data in the form of attributes and code in the form of methods. In object oriented programming, computer programs are designed using the concept of objects that interact with real world. Object oriented programming languages are various but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

Procedural Oriented Programming	Object Oriented Programming
In procedural programming, program is divided into small parts called <i>functions</i> .	In object oriented programming, program is divided into small parts called <i>objects</i> .
Procedural programming follows <i>top down approach</i> .	Object oriented programming follows <i>bottom up approach</i> .
There is no access specifier in procedural programming.	Object oriented programming have access specifiers like private, public, protected etc.
Adding new data and function is not easy.	Adding new data and function is easy.
Procedural programming does not have any proper way for hiding data so it is <i>less secure</i> .	Object oriented programming provides data hiding so it is <i>more secure</i> .
In procedural programming, overloading is not possible.	Overloading is possible in object oriented programming.

In procedural programming, function is more important than data.	In object oriented programming, data is more important than function.
Procedural programming is based on <i>unreal world</i> .	Object oriented programming is based on <i>real world</i> .

---

Q. write differences between procedure oriented languages and oops.

---

### **Java programming**

#### An Overview of Java, Java Environment:

Java language is developed by a team under leadership of James Goslings at SUN Micro Systems (Stanford University Networks). It was actually developed as electronic programming language and called OAK. It was used in set-top boxes, remotes etc. it was released during mid of 1991. Later it is reconstructed for systems and released in the year 1995 with name "java". In French, "JAVA" means "multi-purpose". The java is a general purpose programming language and also it is constructed as a platform (mini operating system).

The general purpose programming edition is called J2SE (Java 2 Standard Edition) and the other platforms released are J2ME (mobile edition) and J2EE (Enterprise Edition). The Java has become more popular because of its features. Its features are generally referred as java BUZZ words.

Object Oriented – In Java, everything is an Object. Java can be easily extended since it is based on the Object model.

Platform Independent – the java program when compiled, it generates byte code. The byte code generated on any OS can be executed on any OS. Because of this byte code the java programs are WORA (Write Once Run Anywhere) OR compile once run anywhere or platform independent.

Simple – As Java syntaxes and semantics are similar to C, C++ etc languages, it is simple to learn and write programs.

Secure – With Java's secure feature it enables to develop virus-free, tamper-free systems. It is managed by the Byte Code Verifier of the JVM.

Portable – as java is platform independent and as the byte code can be interpreted on any platform, it is portable to any system.

Robust – it means healthy. Because of the exception handling mechanism provided by the Java Debugger of JVM, the java programs are healthy and strong.

Multithreaded – With Java's multithreaded feature it is possible to write programs that can perform many tasks simultaneously.

Distributed – Java as it is platform independent it can be distributed to any system on the internet.

Dynamic – Java is considered to be more dynamic than C or C++, as it has the best memory management systems.

---

Q. give a brief introduction to java

Q. write about java features

Q. write about java buzz words

---

Java Environment:

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop java applications and applets. It contains JRE + development tools. When the java was released by SUN micro systems, the software was named J2SDK (Java 2 Standard Development Kit). But later it is called JDK, and the JDK is available in different editions for different purposes. o Standard Edition (general edition for system programming) o Enterprise Edition ( for enterprise web application) o Micro Edition ( for mobile programming)

The JDK when installed, it installs an API and a JVM.

The java API is divided into packages and sub packages that need to be imported to each program to make use of its classes.

The JVM is Java Virtual Machine. The JVM acts as a mini operating system. When we install java into system, we must install the JDK the Java Development Kit. It also installs a public JRE that plugs into the browsers for running java applets. The JDK is designed for each operating system separately.

When we install the JDK into system, the JRE is installed and it includes two components viz., API and JVM. The JVMs are designed separately for each OS and kept within the JDK software. The JVM designed for windows would not work on UNIX or other operating systems. So we must download appropriate JDK and install into system. The “javac” (java compiler) converts high level code into byte code that is JVM dependent. And the JVM further converts the BYTE code into Operating System’s native code. Thus the JVM makes java PLATFORM INDEPENDENT.

The JVM is a software mechanism that includes many layers like byte code verifier, java debugger, interpreter and a compiler called Just In Time compiler.

Byte Code Verifier:

The java compiler when creates byte code file, it places a signature at beginning of the byte code file, called magic number. The byte code verifier of the JVM checks for this magic number and if it exists, then only it allows the byte code to enter into JVM for further process. Thus, the byte code verifier providing a firewall mechanism to provide security to system. Because of this the java is SECURED.

Java Debugger:

The java debugger is the next layer of JVM that checks for logical bugs in the program with the help of a software mechanism called exception handling. This JDB checks for unhandled exceptions OR runtime exceptions. Because of this, the java programs become ROBUST.

Interpreter: as the java applet programs are placed on a web server and executed on client and as the clients can have any operating system, the JVM has an interpreter to convert the byte code into required operating system's matching code called native code where ever , when ever required.

JIT compiler: many a times, the interpreter works slow when compared to compiler, so the JVM is integrated with a compiler called "Just In Time" compiler, that runs by default. It can be switched off, to convert byte code into native code with interpreter.

Thus the JVM converts the byte code into native code for execution by providing security and making java programs robust and platform independent.

---

Q. write about JRE

Q. what are the different components installed when JDK is installed

Q. Explain about JVM

---

### Data types, Variables, scope and life time of variables, and Constants:

#### Data Types in Java:

The data types are used to provide proper treatment to data. The java supports both primitive data types and non-primitive data types (user defined). The different primitive data types supported by java are byte, char, short, int, long, float, double and boolean.

byte: it occupies 1 byte memory in RAM and stores ASCII code of key pressed. It is equivalent to char data type of C.

char: it occupies 2 byte memory in RAM and stores and manages data in Unicode format. (this is not like char data type of C and C++)

short, int and long: these three are used to store and manage integer data. Where the short occupies 2 bytes, int occupies 4 bytes and long occupies 8 bytes in RAM. Though long occupies 8 bytes, it stores only whole numbers only.

float: it occupies 8 bytes memory used to store real numbers. i.e. fractional part also. When we specify a float value in java, the value must be suffixed with a letter F/f to the value. As JRE's default data type for real numbers is double, when we write float x=1.5; it raises error, as 1.5 by default is treated as double. So we must write float x=1.5F; or float x=1.5f;

double: it occupies 16 bytes memory. Used to store real numbers.

boolean: In C and C++ the booleans are managed with PULSE or NOPULSE i.e. with ZERO or ONE, but the java works with booleans with keywords true / false directly. In java, we cannot use 0/1 in logics, we must use the keywords true/false. Ex : boolean b=true;

#### Variable declaration and their scope:

A variable is a named space reserved in main memory, used to store data given input by user. These are used for manipulation of data. The variables can be declared with in a method definition or in a class or as a static variable in a class.

The variables declared with in a method definition are called local variables to that method and they have SCOPE only within that method. They are not accessible in other methods of the class. The local variables of a method cannot be static. And we must initialize them explicitly.

The variables declared with in a class are accessible to all the methods of that class and called instance variables. The instance variables hold the values for each instance of the class. The instance variables have their scope to all the methods of that class.

The static variables are allocated with memory once and shared by all objects. They have common copy in memory. The static variables cannot be declared with in a method. They can be declared only in a class. These will have their scope even if the object is not created.

The variables can be declared to any above data type as it is done in C or C++.

Syntax: data\_type variable\_name;

Ex: int a;

boolean b; etc.

The variables can be assigned with values either at the time of declaration or after declaration.

Ex:

int a;

a=5;

or

int a=5;

Constants :

The java constants are classified into 2 types.

- 1.Constant variables
2. symbolic constants.

The variables declared with in a class with the keyword final is called constant variables in java. Ex: final int x=0;

Once a variable is declared with final keyword and initialised with a value, then that variable cannot be assigned with a new value or cannot be modified with its value. The constants are instance variables.

The variable declared with in a class with both static and final keywords is called a symbolic constant. As these are static these can be accessed with class name directly. These will not be allocated with memory for each instance like constant variables.

---

There can be 5 marks questions on each aspect like... each topic can be written in exam based on question.

Q Write about data type of java

Q. write types of variables and their scope

Q. write about constants in java

---

## Operators:

Operators are used to do process on data. The operators are classified in different ways based on no. of operands (on which operation is performed) they take like unary operators, binary operators and ternary operators. The operators that take one operand are called unary operators. The operators that take two operands are called binary operators. They are further classified into different types based on their purpose.

**Arithmetic Operators :** the operators that are used to perform basic arithmetic operations are called arithmetic operators. They are : +, -, \*, / and %

As the division results two values, the quotient is retrieved with / operator and the remainder with % symbol called modulus. The modulus can be used only with int data type.

**Relational Operators:** these are also called comparison operators. These are used to develop conditions and return true or false, called Boolean values named after the mathematician called George Bool. They are : >, >=, <, <=, ==, !=

The == is called equality operator and the != is not equals to.

**Logical Operators:** these are used to combine two conditions. These are: &&, || and !

The && is logical AND operator, || is logical OR and the ! is NOT. The && results false even if one is false and the || results true even if one is true. The ! operator turns true into false and vice-versa.

**Assignment operator:** this operator assigns RHS to LHS, so LHS must be a variable. It is '='. All arithmetic operators in combination with assignment operator performs dual operations like a=a+2; can be written as a+=2; we can use +=, -=, \*=, /= and %=.

**Increment decrement operators:** these are actually unary operators. They are

++ , - - and -.

The - negates the value.

The ++ increases value by 1 and

the - - decreases value by 1.

These can be used either as pre-process increment/decrement or post process increment/decrement.

**Conditional operator :** the languages C, C++ and java supports an operator '?:' called conditional operator and it is a ternary operator. i.e. it takes three operands.

**Syntax:**

Condition ? statement1 : statement2;

If the condition results true, statement1 is executed by skipping statement2 and vice-versa.

For simpler divisions of program into two parts, we prefer using a conditional operator instead of if-else branching as branching gives compiler overhead (burden on system/RE).

**Special Operators:** Java has some special operators like,

. (DOT) : this is called PERIOD operator. It is member access operator in java. The members of object are accessed through object using this period operator. Ex : obj.feature.

instanceof : this operator is used to check whether the object is of given class type or not.

This operator returns true or false.

new: this operator is used to allocated memory for each instance dynamically.

---

Q. Write about different operators in java.

---

Displaying Output with System.out.println() :

The basic IO operations can be done in java using streams. The output can be displayed onto monitor with the method print() and println() of PrintStream class. The PrintStream class is created with a static object within System class, so we can use System.out.print() or System.out.println() to display data onto monitor. The programmers who are much influenced by syntaxes of C, can also use printf() of System.out object very similar to the printf() of C.

Ex:

```
System.out.printf("%s", "Hello");
```

```
System.out.print("Hello");
```

```
System.out.println("Hello");
```