Programming Language:

In a computer system, we use the programming language to perform a certain kind of operation or we create a process that is executed by the computer system. It is a way of communicating between a human/user/programmer and a computer system.

Procedural Programming Language:

It is a type of programming language where the program or the source code is divided into a small segment which is known as function. In this type of programming language is more emphasis is given on function rather than data. It follows the top-down approach and at the time of developing software, it takes more time. Procedural programming language doesn't support code reusability. Example: C

Object-Oriented Programming Language:

It is a type of programming language that is divided into small segments known as objects. In this programming language, more emphasis is given on data rather than the function. It follows a bottom-up approach. It takes less time for software development. It supports the code reusability mechanism using the inheritance concept. Example: JAVA

Procedural Programming Language vs Object-Oriented Programming Language:

1. In procedural Programming the code is organized around procedures or functions. It follows a linear flow of control where functions are called one after the other to perform tasks. Data is often global or passed explicitly between functions. Object-Oriented Programming is based one the concept of objects, which represent real world entities. It organizes code around objects, each of which contains both data/attributes and methods/functions that operate on that data.

2. In procedural oriented programming language Data and functions are typically separate in procedural programming. Data is often stored in global variables. Object-oriented Programming language promotes encapsulation, which means that data and methods that operate on that data are bundled together in objects. This provides better data hiding and abstraction.

3. Procedural Oriented Programming Reusability is achieved through functions, but it might be limited to the functions themselves. Object-Oriented Programming. emphasizes reusability through the concept of classes and inheritance. You can create new classes by inheriting from existing ones, inheriting their attributes and behaviors.

4. Procedural Oriented Programming Code is organized into functions or modules, but it may lack clear encapsulation. Object-Oriented Programming enforces a higher level of modularity through classes, making it easier to manage and extend the code.

5. Procedural Programming Language Inheritance is not a primary concept in procedural programming. Object-Oriented Programming Language allows for inheritance, enabling the creation of subclasses that inherit attributes and methods from parent classes.

6. Procedural Programming Language Polymorphism is not a built-in feature in procedural languages. Object-Oriented Programming provides polymorphism, allowing objects of different classes to be treated as objects of a common superclass, simplifying code and making it more flexible.

7. In Procedural Programming Language Encapsulation is not as emphasized in procedural programming. Data hiding is often achieved through naming conventions or access control, but it's not enforced by the language.

Object Oriented Programming enforces encapsulation by making data private to objects and providing controlled access through methods.

8. Procedural Oriented Programming Language Suitable for smaller and less complex projects. Object-Oriented Programming Language Better suited for larger, complex applications as it provides a structured way to manage complexity through modularity and encapsulation.

Introduction To Java Programming Language:

Java is a general-purpose high-level programming language. It means using java we can develop verities of applications like desktop application web application enterprise application device application. Java is a technology because java has huge library support for simplifying the code complexity due to support or readymade method. The extension of java is (.java). java is a platform. A platform is an environment where we can execute our java program. java has its own JRE that's why can say java is itself a platform. James Gosling, Patrick Naughton initiated the Java language project in 1991 at Sun Microsystems. This language was initially called "Oak" but in 1995 it was renamed "Java".

Features:

JAVA is compiled as well as an interpreted language. Java supports object-oriented programming. Java is used to develop an internet base of application [applet, servlet, JSP]. Java is used to create dynamic web pages. Java provides facilities to program electronic consumable devices such as mobile, laptops, palmtops, using J2ME. Java supports multithreading.

Java Editions:

In the java programming language there are three types of editions are there.

 # JSE: JAVA STANDARD EDITION: java standard edition provide the basic core functionality of the java programming language. The core java concept is called JSE which is especially used to develop a standalone application or desktop application.

 # JEE: JAVA ENTERPRISE EDITION: java enterprise edition especially used to develop web and enterprise applications. On enterprise platforms, JEE is widely used for developing enterprise-level applications.

 # JME: JAVA MICRO EDITION: java micro edition is used to develop mobile device applications or we can say that android development is the JME of java language.

JDK, JRE, JVM:

JDK: JAVA DEVELOPMENT KIT: this provides an environment to develop and run a java application. So we have to install JDK first.

JRE: JAVA RUNTIME ENVIRONMENT: this provides an environment only to run a java application. Once you install the JDK automatically JRE will create.

JVM: JAVA VIRTUAL MACHINE: JVM is an interpreter who is responsible to run a java program one statement at a time. JVM provides a java execution engine that executes the java source code.

JAVA DEVELOPMENT KIT = JAVA RUNTIME ENVIRONMENT + DEVELOPMENT TOOL

JAVA RUNTIME ENVIRONMENT = JAVA VIRTUAL MACHINE + JAVA LIBRARIES

HOW DOES THE WHOLE PROCESS HAPPEN?

When a programmer writes some lines of source code in java and saves using a specific format like the class name and add an extension (.java). after saving the (.java) file user executes the program in the command prompt of the system using the syntax javac filename.java, and it creates the (.class) file which is a byte code of the java file. JVM will interpret the (.class) file and after that user try to execute the input-output operation of the source code using the syntax java filename and the user will be able to do the input-output operation of the java source code.

First Java Program:

```
class JVP00 {
    public static void main(String args[])
    {
        System.out.println("WELCOME TO JAVA PROGRAM");
    }
}
```

Class JVP00

This line is used the keyword class to declare that a new class name is being defined followed by the class name “JVP00”.

{ = opening curly brace of class

public static void main(String args[])

public is a access specifier used to specify that the source code can be called from anywhere. the main() is declared public because it is called by codes outside the class it is a part of. It is called by the JVM.

Static is a declared because it allows main() to be called without having to instantiate the class. JVM need not create an instance of the class or you can say object of class for calling the main().

Void it does not return a value. the void keyword simply notify to the compiler that the main() does not return anything back to the caller.

String args[] it holds the optional command line arguments passed to the class through the java command line.

{ = opening curly brace of main()

System.out.pritnln(“WELCOME TO JAVA PROGRAM”);

This command is used for print a statement command which print the output and display the output to the user.

In java (;) semicolon is used to end a statement or an operation we use this semicolon numerous time when we write some code in a programming language.

} = end curly brace of main method

} = end curly brace of class

Translator:

When two parties want to share some information between them they prefer communication to share their data or information. But for communication, both parties need a specific language. A language that contains specific symbols which make the communication happen. In humans, they communicate using different types of language to share information/data. But in technology like a computer system they use a number system to communicate to another system.

From starting of the evolution of technology every technology or system uses a binary number system to communicate to another system. But binary number system is very difficult to understand by a human. So to solve this problem we added a translator to every system. A translator defines a process that converts a source language to a specific target language. When we added a translator to a system, the translator converts the binary number system to the specific target language which is understandable by human beings.

There are so many types of translators are there and it differs because of their procedure of translating process. But in a computer system we specifically use three types of translator there are as following.

Compiler:

The compiler is a source code or software which performs translating process in a system and it reads a program written in one language known as source language and converts it into a specific target language. Normally the compiler reads the high-level language which is the source language and it translates into the low-level language which is low-level or machine-level language. So in the procedure of compiler, it read the whole source language at a time for translation and during the translation, it detects the error in the source language and shows the error to the user to solve it. After the conversion of the source to the target language perform the further input-output operation for process execution.

Compile Time Error:

Compile time error are the syntax error which are detected by compiler. They prevent the code from running as it detects some syntax errors. It includes error like missing of semicolon(;), spelling mistake of keywords and identifiers.

Interpreter:

An interpreter is also software that is used for translating but the procedure of the interpreter is different from the compiler. The procedure of the interpreter is it translates each statement of a source code at a time along with it execute the operation of the source language and find the output because it takes one statement for translation at a single clock cycle so the advantage is it detects the error very frequently than the compiler.

Runtime Error:

Runtime error which are not detected by the compiler and produce wrong results. They prevent the code from complete execution. It includes errors such as dividing a number by zero(0), finding square roots of negative number.

Assembler:

Sometimes compiler during translation produces the assembly language as an output so to process the assembly language we use the assembler which converts the assembly language to machine-level language. In assembly language, the production of output is very frequent and easy, and also the debugging process in assembly language is easier than in other languages.