ASSIGNMENT NO : 3

Q1: List down the important features of Java 1.8.

Ans-->

Lambda Expressions −a new feature of the language that allows us to consider actions as objects.

Method References − allow us to define Lambda Expressions by explicitly referring to methods by their names.

Optional − Optionality is expressed via a special wrapper class.

Functional Interface – A Lambda Expression can be used to implement an interface with a maximum of one abstract method.

Default methods − allow us to provide entire implementations in interfaces in addition to abstract methods.

Stream API −a particular iterator class that lets us to efficiently process collections of items

Date API −a Date API inspired by JodaTime that is enhanced and immutable

C	C++	JAVA
It is a procedural language.	It is an object-oriented programming language	It is a pure object-oriented programming language.
It uses the top-down approach.	It uses the bottom-up approach.	It uses the bottom-up approach.
It is a static programming language.	It is also a static programming language.	It is a dynamic programming language.
The code is executed directly.	The code is executed directly.	The code is executed by the JVM.
It also uses a compiler only to translate the code into machine language	It also uses a compiler only to translate the code into machine language	Java uses both compiler and interpreter and it is also known as an interpreted language.
It generates the .exe, and .bak, files.	It generates .exe file.	It generates .class file.
It supports pointer.	It also supports pointer.	Java does not support the pointer concept because of security.
It supports union and structure data types.	It also supports union and structure data types.	It does not supports union and structure data types.
It uses pre-processor directives such as #include, #define, etc.	It uses pre-processor directives such as #include, #define, #header, etc.	It does not use directives but uses packages.
It does not support constructor and destructor.	It supports both constructor and destructor.	It supports constructors only.
It does not support exception handling	It support exception handling	It also support exception handling
It does not support the overloading concept	Method and operator overloading can be achieved.	Only method overloading can be achieved.
It is widely used to develop drivers and operating systems.	It is widely used for system programming.	It is used to develop web applications, mobile applications, and windows applications.

Q2→ diff between C , C++ and java

Q.3→ What is Java?

Ans→

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

Q.4→ What is a package in Java? List down various advantages of packages.

Ans→

A **java package** is a group of similar types of classes, interfaces and sub-packages.Package in java can be categorized in two form, built-in package and user-defined package.There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Advantages of pakages.

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.

2) Java package provides access protection.

3) Java package removes naming collision.

Q.5→ Explain JDK, JRE and JVM?

Ans→

JDK : JDK (Java Development Kit) is a software development kit required to develop applications in Java. When you download JDK, JRE is also downloaded with it.

In addition to JRE, JDK also contains a number of development tools (compilers, JavaDoc, Java Debugger, etc).

JRE : JRE (Java Runtime Environment) is a software package that provides Java class libraries, Java Virtual Machine (JVM), and other components that are required to run Java applications.

JRE is the superset of JVM.

JVM : JVM (Java Virtual Machine) is an abstract machine that enables your computer to run a Java program.

When you run the Java program, Java compiler first compiles your Java code to bytecode. Then, the JVM translates bytecode into native machine code (set of instructions that a computer's CPU executes directly).

Java is a platform-independent language. It's because when you write Java code, it's ultimately written for JVM but not your physical machine (computer). Since JVM executes the Java bytecode which is platform-independent, Java is platform-independent.

Q.6→ Explain public static void main(String args[]) in Java.

Ans→

The main() method is a special method in Java Programming that serves as the externally exposed entrance point by which a Java program can be run. To compile a Java program, you doesn't really need a main() method in your program. But, while execution JVM (Java Virtual Machine) searches for the main() method and starts executing from it.

Public ➔

It is an access specifier that means main() method is accessible globally available . This is necessary because this method is being called by the Java Runtime Environment (JRE) which is not located in your current class. It is important to note that if you make main() method non-public then it's not allowed to be executed by any program, there are some access restrictions applied.

Static ➔

The main() method in Java must be static, because they can then be invoked by the runtime engine without having to instantiate any objects then the code in the body of main() method will do the rest. The main() method should be static because otherwise there would be ambiguity : which constructor should be called? If the main() is allowed to be non-static, then while calling the main() method JVM has to instantiate its class. While instantiating it has to call the constructor of that class, There will be ambiguity if the constructor of that class takes an argument.

void →

Java is a platform independent programming language and if it will return some value then the value may mean different things to different platforms. The "void" is a return type i.e it does not return any value. When the main() method terminates, the java program terminates too. Hence, it doesn't make any sense to return from main() method. If the main() method spawns new threads, then these threads can keep program running. The return type of main doesn't make much sense at this point. If you try to return something from the main method, it will give compilation error as an unexpected return value.

main() →

It's just the name of method or a function name. This name is fixed and as it's called by the JVM as entry point for an application. It's not a keyword.

Q.7 → Why Java is platform independent?

Ans→

Java doesn’t require the entire code to be re-written for all the different platforms. It supports platform independence by using Java byte-code and Java Virtual Machine. Java compiler javac converts the program code into byte code, this byte code is platform-independent and can be run on any operating system's JVM. JVM interprets the byte code to machine code and the program is executed.

Q.8→ **What are wrapper classes in Java?**

Ans→

A Wrapper class is a class which contains the primitive data types (int, char, short, byte, etc)**.** In other words, wrapper classes provide a way to use primitive data types (int**,** char, short, byte, etc) as objects. These wrapper classes come under java.util package**.**

Q.9→ Why pointers are not used in Java?

Ans→

Java do not use pointers because using pointer the memory area can be directly accessed, which is a security issue. pointers need so memory spaces at the runtime. to reduce the usage of memory spaces java does not support pointers.

Q.10→ List some features of Java?

Ans→

Great Performance

The Java compiler is designed for performance. Java code is compiled into bytecode and then compiled by the Java compiler. Post that, it is fed to the JVM (Java Virtual Machine) before it’s converted to machine level code.

Inspired by C and C++

C and C++ are long-tenured programming languages and they are the ancestors of modern programming languages like Java and Python. Java is a bit similar to C and C++ but doesn't have features such as pointers and multiple inheritances. Therefore, having an understanding of C and C++ is useful in learning Java.

Multi-threaded

Multithreading capabilities come built right into the Java language. This means it is possible to build highly interactive and responsive apps with a number of concurrent threads of activity.

Platform Independence

Java has a philosophy called WORA (Writing Once, Run Anywhere). Java code is compiled into an intermediate format, called bytecode, which is to be executed in the JVM (Java Virtual Machine). Any system that runs a JVM is able to execute the Java code.

Truly Object-Oriented

Build upon C++ which is semi object-oriented, Java extends the functionality to become a fully object-oriented programming language. Some of the most important features that make it an object-oriented puritan are:

Abstraction

Encapsulation

Inheritance

Polymorphism

Robust

Java guides the programmer to adopt important programming habits required for the creation of highly reliable applications. Unlike C and C++, Java relies on a simple memory management model reinforced by the automatic garbage collection feature.

Secure

Safety features are built into the language and runtime systems. These include runtime checking and static type-checking at compile time. With such features in place, it becomes a daunting task to invade a Java application from the outside.

Simple

Ease of reading and writing makes any language simple. This holds true for Java as it has a less ambiguous syntax terminology. Anyone can start right off with Java with an understanding of the basic underlying principles of programming.

Q.11→ Why is Java Architectural Neutral?

Ans→ Java is architecture neutral because there are no implementation dependent features, for example, the size of primitive types is fixed. In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture.

Q.12→ How Java enabled High Performance?

Ans→ Java enabled High performance by introducing JIT- Just In Time compiler , JIT helps the compiler to compile the code On demand basis i.e which ever method is called only that method block will get compiled making compilation fast n time-efficient. This makes the java delivering high performance.

Q.13→ Why Java is considered dynamic?

Ans→Java is considered dynamic because of Bytecode. The source code which is written in one platform that code can be executed in any platform. It loads the class file during runtime only. Hence, any thing that happens in runtime is dynamic.

Q.14→

Java Virtual Machine (JVM) is a specification that provides runtime environment in which java bytecode (.class files) can be executed. The JVM is the platform. As the name implies, the JVM acts as a"virtual" machine or processor. Java's platform independence consists mostly of its Java Virtual Machine (JVM) . JVM makes this possible because it is aware of the specific instruction lengths and other particularities of the platform (Operating System).

The JVM is not platform independent. Java Virtual Machine (JVM) provides the environment to execute the java file(. Class file). So at the end it's depends on your kernel , and kernel is differ from OS (Operating System) to OS. The JVM is used to both translate the bytecode into the machine language for a particular computer, and actually execute the corresponding machine-language instructions as well. Without the JVM, you can’t run a Java application.

Q.15→ List two Java IDE’s?

Ans→ Eclipse & IntelliJidea