Page No.
Date

H.W. (Day 7)

1) Identify the feautures for the following versions :- Java 8, 15, 17, 19.

→ 1) Java 8

- Oracle released a new version of Java as ~~Java~~ Java 8 in March 18, 2014. It includes various upgrades to the Java programming, JVM, Tools and Libraries.
- Java 8 provides following feautures :-
- Lamda expressions
- Method references.
- Functional interfaces
- Stream API
- Default methods
- Base64 Encode Decode
- Static methods in interface.
- Optional class
- ForEach() method
- Nashorn JavaScript Engine
- Parallel Array Sorting
- Type and Repating Annotations
- Concurrency Enhancements
- JDBC Enhancements

Page No.
Date

2) Java 15
- It was released on 16th September 2020

- Java 15 Provides following Features :-
- Sealed Classes
- Pattern Matching for instance-of
- Records
- Text Blocks
- Hidden classes
- Remove the Nashorn JavaScript Engine
- Reimplement the Legacy DatagramSocket
- Disable and Deprecate Biased Locking
- Shenandoah : A Low - Pause - Time Garbage collector
- Remove the Solaris and SPARC Ports
- Forign - Memory Access API
- Deprecate RMI Activation for Removal.

Page No.
Date

3) Java 17
- It was released on 14th September 2021

- Java 17 Provides following features
- Restore or Rebuild the "Always - Strict Floating-Point" Semantics.
- Enhanced faster "pseudo - Random" Number Generat
- New macOS rendering pipelines.
- macOS / AArch64 Port.
- Dismiss the Applet API for Removal
- JDK Internals Encapsulates Strongly
- Switch Pattern Matching
- Activation of the Removal RMI
- Generate Sealed classes.
- Removal of the Experimental AOT & JIT compiler
- Remove the Security Manager.
- Foreign Functions & Memory API (Incubator)
- Vector API (Second Incubator)
- Deserialization Filters Based on Context.

Page No.
Date

4) Java 19
- It was released on 20th September 2022
- Java19 Provides following features.
• Support Unicode 14.0
• New system properties for System.out & System.err
• HTTPS Channel Binding Support for Java GSS/Kerberos
• Additional Date - Time formats.
• New Methods to create Preallocated HashMaps & HashSets.
• Support for PAC - RET Protection on Linux/AArch64
• Automatic Generation of the CDS Archine
• Windows KeyStore Updated to Include Access to the local Machine Location
• TLS signature Schemes
• Add a provider path option to jarsigner

Page No.
Date

2) Comparision of C++ & Java

→

C++	Java
- C++ is platform dependent	- Java is platform independent.
- C++ is mainly used for system programming	- Java is mainly used for application programming.
- C++ was designed for systems and applications programming. It was an extension of the C programming language.	- Java was designed and created as an interpreter for printing systems but later extended as a support network computing. It was designed to be easy to use and accesible to a broader audience.
- C++ supports the goto statement.	- Java doesn't support the goto statement.
- C++ supports multiple inheritance.	- Java doesn't support multiple inheritance through class. It can be achieved by using interfaces in java.
- C++ supports operator overloading.	- Java doesn't support operator overloading.
- C++ supports pointers you can write a pointer program in C++	- Java supports pointer internally. However, you can't write the pointer program in java. It means java has restricted pointer support in java.

- C++ uses compiler only. C++ is compiled and run using the compiler which converts source code into machine code so, C++ is platform dependent.	- Java uses both compiler & interpreter. Java source code is converted into bytecode at compilation time. The interpreter executes this bytecode at runtime and produces output. Java is interpreted that is why it is platform independent.
- C++ supports both call by value and call by reference.	- Java supports call by value only. There is no call by reference in java.
- C++ supports structures & unions.	- Java doesn't support structures & unions.
- C++ doesn't have built-in support for threads. It relies on third-party libraries for thread support.	- Java has built-in thread support.
- C++ doesn't support documentation comments.	- Java supports documentation comment (/** ... */) to create documentation for java source code.
- C++ supports virtual keyword.	- Java has no virtual keyword.

- C++ doesn't support >>> (Unsigned right shift) operator.	- Java supports (Unsigned right shift) >>> operator.
- C++ always creates a new inheritance tree.	- Java always uses a single inheritance tree because all classes are the child of the object class in java. The Object class is the root of the inheritance tree in java.
- C++ is nearer to hardware.	- Java is not so interactive with hardware.
- In C++, a single root hierarchy is not possible	- Single root hierarchy is possible as everything gets derived from java.lang.Object.