ASSIGNMENT-1

Page No.
Date

1] Java Development kit (JDK)

→ JKD is a bundle of Software development tools and supporting libraries Combined with the Java Runtime Environment (JRE) & Java virtual Machine (JVM)

i) Java Virtual Machine :-

- JVM is a Software tool responsible for Creating a run-time environment for the Java Source Code to run.
- The very powerful feature of Java is "Write once & run anywhere" it is made possible by JVM.
- The JVM Stays right on top of the host operating System and Converts the given Java Source Code into Bytecode (Machine language) & executes the program

ii) Java Run-time Environment (JRE) :-

- JRE is a Software platform where all the Java Source Codes are executed.
- JRE is responcible for integrating the Software plugins, Jar files, and Support libraries necessary for the Source Code to run

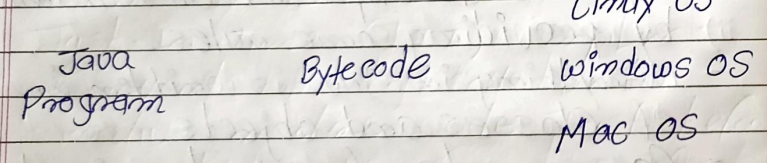
- Components of JDK in Java :-
- Java - It acts as the deployment launcher in the older SUN Java. It loads the Class files and interprets the Source Code Compiled by the Javac Compiler
- Javac - The Javac Specifies the Java Compiler to convert the Source Code into byteCode.
- Javadoc - the Javadoc generates documentation for the Comments added in the Source Code
- Jar - The Jar helps the archives to manage the Jar files in the package library.

Q2] List and explain the Salient features of Java

→ i) Simple :-
- Java is designed to be easy to learn
- Its syntax is quite simple, clean and easy to understand
- Java is inspired by C and C++
- There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java

ii) Object Oriented :-
- In Java, everything is an object which has some data and behaviour.
- Java can be easily extended as it is based on object model.
- OOPS is a methodology that simplifies Software development and maintaince by providing some rules.
- Everything in Java is written in terms of Classes and objects
- Components of object oriented programming :
  - Object
  - Class
  - Inheritance
  - Polymorphism
  - Abstraction
  - Encapsulation

iii) Platform Independent :-
- Java is Write once Run anywhere language
- A platform is the hardware or software environment in which a program runs
- Java provides a software based platform
- Java Code can be executed on multiple platforms like widows, Linux, Sun Solaries, macos etc.
- On Compilation Java program is Compiled into bytecode. This bytecode is platform independant and can be run on any machine, plus this bytecode format also provide Security.
- Any machine with JRE can run Java program



iv) Architecture :- Neutral
- Java Compiler generates an architecture-neutral object file format, which makes the Compiled Code executable on many processors, with the presence of Java Runtime System.

v) Portable :-
- Being architecture-Neutral and having no implementation dependent aspects of the Specification makes Java portable. Everything related to Storage is Predefined.

vi) Robust
- Java uses Strong Memory Management

Q.3] List and Explain the Components of Java Virtual Machine

→ i) Class loader :-
- Class loader is a subsystem of JVM which is used to load Class files.
- whenever the Java source file is Compiled it is Converted into bytecode as a "Class" file
- When this Class is used, the class loader loads it into the main memory.
- the first class to be loaded into memory is usually the class that Contains the main() method.
- This are three phases
  - Loader : this phase loads files from Secondary memory into the main memory for execution
    loading envolves taking the binary representation of a class or interface with a particular name and generating the Original Class or interface from that.

- Linking : Linking is a Class or interface involves Combining the different Elements and the dependences of program together. Linking include
- Verification :- This phase checks the Structural Correctness of the "Class" file by Checking.
- Bytecode verifier - Ensures that the byte code generated by the Java Compiler addons to the Java language & Specification & does not vilate Security Constrains. It helps to prevent Certain Security Unknerabilities.
- Interpreter - The bytecode line by line & executes it while. this approch is Straight forword. It Can be run efficient Compared other execations methods.
- Just-In-Time (JIT) Compiler - Converts the bytecode into native machine Code Just before execution. This Can Significantly improve performance or native machine Code is typically faster then interpreted byte code.

Q.4] Write in detail about different types of operation in Java. Category wize quoting their functionality, operands & return type. Give one example statement of each.

→ Certainly in Java operations Can be Categorized based on their functionaly Here are Some Common type :-

- Arithmatic Operations :-
  functionality - Perform basic Arithmatic operations.
  Operation - Numeric values.
  Return type - Same as the operands.
  Example -
  ```
  int result = 10 + 5 ; //Addition.
  ```

- Relational Operator :-
  functionality - Compare value & return boolean value.
  Operation - Any primative data type
  Return type - Boolean
  Example -
  ```
  BooleanEqual (7 == 7);
  ```

- Logical operator :-
  functionality - Assign value the variable
  operator - Variables & values.
  Return type - Same as the assign value.

Example - int re = 1 ;

- Increment & Decrement operator :
functionality - Increase or Decrease the value of variable
operands - variables
Return type - Same as the variable type
Example - i++ ; // Increment.

Q.5] What are the Primitive data type in Java ? Briefly explain their Size range & other details.

→

	SIZE	RANGE
Byte	8 bit	128 to 127
Short	16 bit	32768 to 37767
int	32 bit	2731 to 2731-1
long	64 bit	2^{m63} to 2^{63-1}
float	32 bit	Single-Precision floating point
double	64 bit	Double-Precision floating point
Char	16 bit	Unicode Character.

Q.6] Explain about Memory Management in Java with reference to Stack & heap

→ In Java memory management involves the allocation & deallocation of memory for object during program execution the memory is divided into two main Area which are Stack & Heap.

- Stack =
i) Purpose - The Stack is used for storing local variable & managing invocations
ii) Size & Allocation - Memory allocation is automatic & follows a last in first out (LIFO) structure. Each thread has its own Stack & the size is usually smaller compared to the heap.
iii) Data types - Store Primitive data type & reference to object.
iv) Lifetime - Short lived memory is automatically reclaimed when the method execution completes.

- Heap =
i) Purpose - The heap is used for dynamic memory allocation primarily for objects & arrays.

ii) Size & Allocation - Memory allocation is managed by the Java Virtual machine (JVM).
The heap size can be adjusted using JVM parameters.

iii) Data types - Store objects & arrays object have a longer lifetime & may exists beyond the scope of a single method.

iv) Memory leaks - If reference to objects are not properly managed memory leaks can occure, impacting performance.

Q.7] Explain the term Narrowing Widening

→ In Java narrowing & widening refers to type conversion between different data types specifically concerning numeric type.

i) Widening -
Widening is also known as implicit conversions, occurs when a value of a smaller data type is automatically converted to a larger data type.
Example, Converting int an 'int' to a 'long' OR float to a double.

ii) Automatic - It happens automatically here generally no loss of precision because the largest type can represent the entire range of the source type.

iii) Narrowing - (Explicit Conversion)
Narrowing or Explicit conversion occurs when a value of a larger data type is explicitely converted to a smaller data type.
Example, Converting 'double' to an int OR float to short.

iv) Manual Casting -
It requires manual intervention through casting & there might be loss of precision if the target type cannot represent.

Q.8] Write in details about Static Keyword.

→ In Java static keyword is used to declare members that belong to the class rather then instance of the class. It can be applied to variables methods insted classes & blocks. Here are detailed explanation about how the static keyword used in various contexts.

i) Static Variables -
Variables declared with the Static keyword are known as Static variables. or class variables.
Scope - They are shared by all instance of the class & belong to the Class rather than individual object.
Access - Accessed using the Class rather than an instance

ii) Static Method -
Methods declared with the Static key word are called as Static Methods.
Access - Class used the class name rather than an istance. They cannot access non-static members. directly.
Example - Commonly used for the utility method or operation that don't on specific instance State.

Q.9] Write a short Note on Access Specifiers in Java.
→ Access Specifiers in Java determine the visibility & Accessibility of classes, methods & variables in a program, there are four access specifiers in Java.

i) Java Public -
The most permissive access level public members are accessible from any other Class.
Example -
```
Public Class Example {
    public int publicVariable;
    public void publicMethod() {
    }
}
```

ii) Protected -
Accessible within the same Package & Subclasses even if they a in different packages.
Example -
```
Class Example {
    protected int protectedVariables
    protected void protectedMethod
        // Code here
    }
}
```

iii) Default -
If there is an no Access Specifier is specified the default access level is package. private members are accessible only within the same package.

Example:-

```
Class Example {
    int defaultVariables;
    void defaultMethod() {
        // Code here.
    }
}
```