Q.2 list and explain the Salient features of Java.

→ The Primary objective of Java programming language creation was to make it portable simple and secure programming language. The features of Java are also known as Java buzzwords.

A list of the most important features of the Java language is given below.

1, Simple :

Java is very easy to learn, and its syntax is simple, clean and easy to understand.

2, object oriented :

Java is an object oriented programming language. Everything in Java is an object.

- Basic concept of oops are: object, class, inheritance, polymorphism abstraction. Encapsulation.

3, Portable

Java is portable because it facilities you to carry the Java byte code to any platform. It doesn't require any implementation

4) Platform independent:
Java is platform independent because it is different from other language like C, C++, etc

5) Secured:
Java is best known for its Security with Java we can develop virus free System.

6) Robust -
Java is robust because It us uses Strong memory management

7) High performance -
Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code

8) Dynamic:
Java is Dynamic language It means Classes are loaded on demand

Q.3 List and Explain the component of Java virtual machine

-> The Java virtual machine (JVM) is a component of the Runtime. Environment (JRE) and playing a central role in executing Java bytecode Here are the

⑤ Memory Area:

- Divided into Several Section include Heap Manage memory for object and is shared among all thread
- Stack : Stores local variable and method call information for each threads.
- Method Area : Stores class level data method information and static variables

⑥ Execute Engine :

consists of the interpreter and the JIT compiler It execute The Java JIT bytecode either by interpreting it or compiling it to native machine code for improved performance These components work together to provide a platform independent execute environment for Java program

Q.4 write in details about different types of operation in Java, category wise covering functionality: approach and return type given an example statement for each.

-> Certainly In Java operators can be categorized based on their functional Here are some common type.

Q7) Explain the term - ~~narrow~~ narrowing widening

-> In Java narrowing and widening refer to type conversion between different data types specifically concerning numeric type.

i) widening :-

Definition :- widening also known as implicit conversion occurs when a value of a smaller data type is automatically converted to larger data type :

Exple :- converting an 'int' to a 'long' or a float to a double

ii) Automatic :- It happens automatically a There generally no loss of precision because the target type can repsent the entire range of the same type

iii) Narrowing (Explicit conversion)

Defination :- Narrowing or explicit conversion occurs when a value of a larger data type is explicitly converted a smaller data type.

Example :- converting 'double' to an int or a float to a short

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

- Size and Allocation:- Memory allocation is automatic and follows a last into first out (LIFO) structure Each Tread has its own stack and the size is usually smaller compared to the heap.
- Data Type:- Store primitive data type and reference to objects
- Lifetime: short-lived memory is automatically reclaimed when the method execution completed.

ii) Heap:-

- Purpose: The heap is used for dynamic memory allocation primarily for objects and arrays.
- Size and Allocation: memory allocation is managed by the Java virtual machine (JVM) the heap is size can be adjusted using JVM parameters
- Data Types: Store objects and arrays object have a longer lifetime and may exist beyond the scope of a single method.
- Memory Leaks: If reference to objects are not properly managed memory leaks can occur, impacting performance
-

Q.8 Write in detail about static Keyw

-> In Java the static keyword is used to declare member that belong to the class rather than instance of the class. It can be applied to variable method, nested classes and blocks. Here are detailed explanation to how the 'static' keyword used in various contents.

i) Static variablq:-

- Defination · variables declared with the Static keyword are known as Static variable or class variables
- Scope: They are shared by all instance of the class and belong to the class rather than individual object.
- Access: Accessed using the class rather than an instance.

ii) Static method:-

- Definition :- Methods declared with the static Keyword' Static method.
- Access - claim used the class name rather than an instance. They cannot access non-static members directly.
- Example :- commonly used for utility method or operation that don't on Specific instance State

Q9 Write a short note on access Specifier in Java.

→ Access specifier in Java determine the visibility and accessibility of classes, methods and variable in a program. There are four access Specifier in Java

i) Java public:-

Description: The most permissive access level public members are accessible from any other class.

Example:-

```
public class Example{
  public int public variable;
    public void public Method (){
  }
  }
```

ii) protected :-

Description:- Accessible within the same package and subclass even if they a in different packages.

```
class Example{
  protecte int protected variable
    protecte void protected method
          // code here
  }
```

iii) Default :-

Description:- If no access specifier is specified the default access level is package-priv
Member are accessible only with in

Same package.

Example

```
class Example
   int default variable;
      void default Method (){
         // code here
   }
}
```

Q5) what are the primitive datatype in Java.? briefly explain their size range and other details

→	size	range
1) byte	8 bit	128 to 127
2) short	16 bit	32768 to 32717
3) int	32 bit	2^{31} to 2^{31-1}
4) long	64 bit	2^{63} to 2^{63-1}
5) float	32 bit	Single - precision floating point
6) double	64 bit	double - precision floating point
7) Char	16-bit	unicode character

Q-6 Explain about memory management in Java with reference to stack and heap.

→ In Java memory management involves The allocation and deallocation of memory for object during program execution the memory is divided into two main area the stack and the heap

i) Stack :-

purpose: - The Stack is used for storin local variable and managing invocation.

main components of the Java virtual machine

① class loader:

Responsible for loading classes needed by the Java program during runtime. It takes the compiled Java classes and make them available for the JVM.

② Bytecode verifier:

Ensure that the bytecode generated by the Java language compiler adder. to the Java language Specification and does not violate security constr. It help prevent con certain security vu vulnerabilites.

③ Interpreter

Interprates the bytecode line by line and executes it while this approach is straight forward. It can be less efficient compared other execution method.

④ Just-In-Time (JIT) compiler:

Converts the bytecode into native machine code Just before execution This can Significantly improve as native machine code is typically faster than interpreted byte code

- Arithmetic operation:
- Functionality: perform basic Arithmatic open
- operation:- Numeric value
- Return type:- same as the operand
  Example:-
  int result= 10+5; // Addition
- Relational operator:-
- functionality : compare value and return a boolean result
- operation - Any primitive data type
- Return Type:- boolean
  Example: boolean Equal (7==7);
- logical operator:
  functionality: perform logical operation a: boolean value
  - operator: boolean value
  - Return value : boolean
- Assignment operator:-
- functionality : Assign value the variable
- operator: variable and values
- Return type: Same as the assign value
  Example : int x=1;
- Increment & decrement operation
- functionality : Increase or decrease the value of a variable
- operand: variable
- Return type : Same as the variable type