

Q.1 Write a short note on Java Development Kit?

- i) Java Development Kit (JDK) is a software development kit used for Java Programming.
- ii) It includes tools executable and binaries required for Java application developer.
- iii) JDK comprises the Java Runtime Environment (JRE) executable libraries and development tools like compilers and debuggers.
- iv) Developers use JDK to create compile and run Java application.
- v) It plays a crucial role in building in robust platform and platform independent software machine is a fundamental component for Java developers.
- vi) JDK contains:-
 - Java Runtime Environment (JRE)
 - An interpreter loader (Java)
 - A compiler Java ()
 - An archiver (jar) and many more
- vii) You can use the JDK compiler to convert your Java text file into an executable program.

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 easy to learn language because
the features of Java are also simple,
so Java is known as

A list of the most important features of
the given language is given below:-

Simple:-

Java is very easy to learn and easy to
understand.

i) Object oriented:-

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

Basic concepts of OOPS are:-

Object, class, Inheritance, Polymorphism

Abstraction, Encapsulation.

ii) Portable:-

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

iv) Platform independent:-

Java is platform independent because
it is different from other language like
C, C++, etc.

v) Secured:-

Java is best known for its security

with Java we can develop virus free systems

vii) Robust:-

Java is robust because it uses strong memory management.

viii) High Performance:-

Java is faster than the other independent programming language because Java bytecode is close to native code.

viii) Dynamics:-

Java is dynamic language. It means classes are loaded or demand.

Q.5] List and explain the components of Java Virtual Machine.

→ The Java Virtual Machine (JVM) is a component of the Java Runtime Environment (JRE) plays a central role in executing Java bytecode. Here are the main components of the Java Virtual Machine.

i) Class Loader:-

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

ii) Bytecode verifier:-

Ensure that the bytecode generates by the Java Compiler address to the Java language specification and does not violate security.

constraint. It helps prevent certain security vulnerabilities.

iii) Interpreter:-

Interpreter the bytecode line by line and executes it while. This approach is straightforward. It can be efficient compared to other execution methods.

iv) Just-In-Time (JIT) compiler:-

Converts the bytecode line by line and executes. This can significantly improve performance as native machine code is typically faster than interpreted byte code.

v) Memory Area:-

- Divided into several sections include heap which manage memory for objects and is shared among all threads.
- Stack:- Stores local variables and method call information and static variables.

vi) Execute Engines:-

Consists of the interpreter and the JIT compiler. It executes the Java JIT bytecode either by interpreting or compiling to native machine code for improved performance.

Q.4) Write in details about different types of operation in Java, category wise quoting functionality approach and return type. Give an example statement for each.

→ Certainly in Java operators can be categorized based on their functionality.

Here are some common types:

• Arithmetic operation:-

Perform basic Arithmetic oper.

Operation:- Numeric Values

Return type:- same as the operands

Example:-

```
int result = 10 + 5 //Addition
```

• Relational Operator:-

Functionality:- Compare value and return a boolean result

Operation:- Any primitive data type

• Logical Operator:

Functionality :- Perform logical operation on boolean values.

Return type:- Return true if 1 is condition

• Assignment Operator:-

Functionality:- Perform logical operator on boolean values.

Return type :- Same as the variable

Example:- int x = 1

Q5) What are the primitive datatype in Java?
Explain their size range and other details.

	Size	Range
1 byte	8 bit	-128 to 127
2 short	16 bit	-32768 to 32767
3 int	32 bit	-2147483648 to 2147483647
4 long	64 bit	-9223372036854775808 to 9223372036854775807
5 float	32 bit	Single precision float point
6 double	64 bit	double precision float point
7 char	16 bit	unicode character

Q6) Explain above memory management in Java with reference to stack and heap.

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

If stack:-

- Purpose:- The stack is used for storing local variables and managing invocations.
- Size and Allocation:- Memory allocation is automatic and follows a last in first

Ort(LIFO) structure each thread has its own stack and the size is ~~essentially~~ usually smaller compared to the heap.

Datatypes:-

Store primitive datatypes and reference to objects.

Lifetime :- Short lived memory is automatically reclaimed when the method execution completes.

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 size can be adjusted using JVM parameters.

DataTypes:-

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.

Q7 Explain the terms Narrowing: Widening.

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

i) Widening:-

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

ii) Automatic:

It happens automatically 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 explicitly converted to 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. There might be loss of precision if the target type cannot represent.

Q8 Write

→ In Java declare variable. It can have explicit user defined static variable or by local variable.

Var key or by local variable.

Ans or

Q8) Write a detail about static keyword.

→ In Java the static keyword is used to declare members that belong to the class rather than instance of the class. It can be applied to variables, methods, nested classes and blocks. Here are detailed explanations to how the 'static' keyword is 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 instances of the class and belong to the class rather than individual object. Accessed using the class rather than an instance.

ii) Static Method:-

Method declares with the static keyword's static methods.

Class 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.

Q.9] Write a short note on access specifier in Java.

iii) Default

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

ij) 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() {

}

}

ij) Protected:-

Accessible within the same package and subclasses even if they in different packages.

Example:-

class Example {

 protected int protectedVariables;

 protected void protectedMethod()

{}

iii) Default:

If no access specifiers is specified
The default access level is package.
private members are accessible only
with in same or some package

Example:-

```
class Example{  
    int default variables;  
    void defaultMethod () {  
        }  
    }  
}
```