1/2/24

Assignment - 1

Q.1] Write a short note on Java Development kit.

→ The Java development kit (JDK) is a cross-platformed software development environment that offers a collection of tools and libraries necessary for developing Java-based software applications and applets. It is a core package used in Java, along with the Java virtual machine and Java Runtime.

- The JDK has a private java Virtual machine (JVM) and a few other resources necessary for the development of a Java Application.

JDK contains :-

- Java Runtime Environment (JRE)
- An interpreter / loader (JAVA)
- A compiler (JAVAC)
- An archiver (jar) and many more.

- The Java Runtime Environment in JDK is usually called Private Runtime because it is separated from the Regular JRE and has extra content. The Private runtime in JDK contains a JVM and all the class libraries present in the production environment, as well as additional libraries useful to developers, eg. internationalization libraries and the IDL libraries. and the IDL libraries.

Most popular JDKs:-

- Oracle JDK, OPEN JDK, Azul System Zing, Azul System, IBM J9 JDK and Amazon coretto.

PAGE NO.
DATE / /

- Compile and Run Java code using JDK
- The JDK compiler is used to convert the Java text file into an executable program. The Java text segment is converted into bytecode after compilation which carries the class extention.

- Important Components of JDK

Component	Use.
javac	Java compiler converts Source code into byte code.
java	The loder of the java apps.
javap	Class file disassembler
javadoc	Documentation generator.
jar	Java Archiver helps manage JAR files.
appletviewer	Debugging of JAVA applets without web browser.
xjc	Accepts XML schemes and generate Java c
apt	Annotation - procesing tool.
jdb	Debugger
imc	Java Mission Control.
iConsole	Monitoring and Management Console
pack200	JAR compression tool.
extcheck	Utility tool to detect JAR file conflicts.
idlj	IDL to Java compiler.
keytool	The keystore manipulating tool.
jstatd	jstat daemon (experimental)
jstat	JVM statistic monitoring tool.
jshell	jshell introduced in java 9.

Q.2) List and explain the salient features of JAVA.

→

The salient features of JAVA are:-

(1) Object Oriented

- It means that the software is organize as a combination of different types of objects.
- It is a methodology that simplifies the software development and maintainance by providing some rules. The Basic concepts of Oops are object, class, inheritance, polymorphism, Abstraction & encapsulation.

(2) Simple

- Java is very easy to learn and its syntax is simple clean and easy to understand. Java has removed many complicated and rarely-used features, Ex- explicit pointers, operator overloading, etc.

(3) Secured

- Java is best known for its Security. With Java we can develop virus-free systems. Java is secured because of No explicit pointer, Java programs run inside a virtual machine, sandbox, classloader, Bytecode Verifier, Security Manager.

(4) Platform independent

- Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platforms specific machine while java is a write once, run anywhere language.

5 Robust (strong)

- Java is robust because :-
- It uses strong memory management
- There is a lack of pointers that avoids security problems.
- Java provides automatic garbage collection which runs on the Java Virtual machine to get rid of objects which are not being used by a Java application anymore.
- There are exception, handling and the type checking mechanism in Java. All these points make java robust.

6 Portable

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

7 Architecture-neutral

- Java is architectural neutral because there are no implementation dependent features, for ex:- the size of primitive types is fixed.

8 Dynamic

- Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports function from its native languages i.e C and C++. Java supports dynamic compilation and automation memory management. (garbage collection).

⑨ Multi-Threaded

- A thread is like a Separate program, executing concurrently. Java Programs that deal with many tasks at once by definning multiple threads. The main advantages of multi-threading is that it doesn't occupy memory for each thread.

⑩ Distributed

- Java is distributed because it facilitates users to create distributed applications in Java. RMI and EJB are used for creating distributed applications

⑪ High-Performance

- Java is faster than other traditional interpreted programming languages because java bytecode is close to native code. It is still a little bit slower than a compiled language. Java is an interpreted language that is why it is slower than compiled languages.

PAGE NO.
DATE: / /

Q.3] List and explain the components of Java Virtual machine

→

Java (Java Virtual Machine) is an abtract machine. It is a Specification that provides runti environment in which java bytecode can be executed. JVM are available for many hardware and software platforms.

The Components of Java Virtual Machine are :-

① Classloader :
- Classloader is a subsytem of JVM which is used- classload files. Whenever we run the java program it is loaded by the classloader. There are three built-in classloaders in Java
① Bootstrap classloader.
② Extension classloader
③ Application clasloader.

② Class (Method) Area :
- Class (Method) Area stores per-class structure such as the runtime constant pool, field and method data the code for methods.

③ Heap
- It is the runtime data area in which objects are allocated.

④ Stack
- Java stack stores frames. It holds local variables and partial results, and plays a part in method

invocation and return. A new frame is created each time a method is invoked. A frame is destroyed when its method invocation completes.

(5) Program Counter Register

- PC (Program counter) register contains the address of the java virtual machine instruction currently being executed.

(6) Native Method Stack

- It contains all the native methods used in the application.

(7) Execution Engine.

- It contains a virtual processor, interpreter and Just-In-Time Compiler. It is used to improve the performance. JIT compiles parts of the byte code that have similar functionality at the same time, and hence reduces the amount of time needed for compilation.

(8) Java Native Interface

- Java Native Interface is a framework which provides an interface to communicate with another application written in another language. Java uses JNI framework to send output to the console or interaction with OS libraries.

Q.4) Write in detail about different types of operato in java, category wise quoting their functionalit operands and return type. Give one example o statement for each.

→

- Operator in Java is a Symbol that is used to perform operations.
- There are different types of operators in Java:-

1. Unary Operator
2. Arithmetic Operator
3. Shift Operator.
4. Relational Operator
5. Bitwise Operator
6. Logical Operator
7. Ternary Operator.
8. Assignment Operator.

1. Jara Unary Operator

- The Jara Unary Operators require only one operand Unary operators are used to perform various operations i.e. incrementing / decrementing, negating an expression & inverting the value of a boolean.

Operator type	Category	Precedence
Unary	Postfix	expr++ expr--
	Prefix	++expr --expr +expr -expr

Ex:-
```
Class Operators
{
 public static void main(string args[])
 {
 int x=10;
System.out.println(x++);
System.out.println(--x); } }
```

=> Output :-

```
11, 10
```

② Arithmetic Operators :-

- Java arithmetic operators are used to perform addition, subtraction, multiplication and division.

Operator type	Category	Precedence.
Arithmetic	multiplicative	* / %
	additive.	+ -

Ex:-
```
class Operator
{
 public static void main (String args[])
{ int a = 10;
  int b = 20;
System.out.println (a+b); >> 30
System.out.println (a-b); >> -10.
System.out.println (a*b); >> 200
System.out.println (a/b); >> 1/2
System.out.println (a%b); >> 0.5
}}
```

③ Shift Operator.

- There are two shift Operator :- Its

i) Left → It's shift all of the bits in a value to the left shide.

ii) Right → It's shift all of the bits in a value to the right side.

Operator type	Category	Precedence
Shift Operator	-shift	<< >> >>>

Ex:-
```
class Operator
{
 public static void main (String , args[])
 {
  System.out.println (10<<2) >>40
  Sytem.out.println (10>>2) >>2
 }
}
```

④ Java Ternary Operator

- Java Ternary Operator is used as one time replacement for if-then-else statement and it is only conditional operator which takes three operands.

Operator type	Category	Precedence
ternary operator	Ternary	?:

Ex:-
```
class Operator
{
 public static void main (String args [])
{int a=2, b=5;
 int min;
 min =(a<b)? a:b;
 System.out.println (min);
}
}
```

(5) Assignment Operator

- Assignment Operator is one of the most used to assign the value on its right to the operand on its left

Operator type	Category	Precedence
Assignment operator	Assignment	=, +=, -=, *=, /=, %=, &= ^=, |=, <<=, >>=, >>>=

Ex:-

```
class Operator
{
  public static void main (String args[])
  {
    int b = 20; c= 2 ; d = 4 ;
     int  a =10 ;
a+ = 4 ;
b - = 4 ;
 c *= 2 ;
 d / = 2 ;
System.out.println("a")  >> 14
System.out.println ("b")  >> 16
System.out.println ("c")  >> 4
System.out.println ("d")  >> 8
}}
```

(6) Relational Operator

- In Relational Operator ; comparison operator is used to compare the two values.

Operator type	Category	Precedence
Relational operator	comparison	< > <= >=

⑦ Bitwise & logical Operators:-

Operator type	category	Precedence
Bitwise.	AND, exOr, in, or	&, ^, |
Logical	AND OR	&&, ||

Logical && and bitwise &&

- ① The logical && operator doesn't check the Second condition if the first condition is false it checks the second condition.

② The bitwise & operator always checks both condition whether first condition is true or false.

Logical || and bitwise |

- ① The logical operator doesn't check the Second condition if the first condition is true. It checks the second condition only if the first one is false.

② The bitwise | operator always checks both conditions whether first condition is true or false

Q.5) What are the primitive data types in Java? Explain their size and range · other details too:

→

Primitive data types are the building blocks of data manipulation. There are the 8 types of Primitive data types they include:-

① boolean data type
② Byte datatype
③ char datatype.
④ Short datatype.
⑤ int datatype.
⑥ long datatype
⑦ double datatype.

① Boolean Datatype.

- The boolean data type is used to store only two possible values true & false. This data type is used for simple flags that track true/false condition. The size is 1 byte.

② Byte datatype.

- Byte data type is an 8 bit signed two's complement integer. It's value-range lies between -128 to 127. It includes -128 minimum values and 127 maximum and default is 0. The size is 1 byte.

③ Char:- The char datatype is a Single 16-bit unicode character. It values range lies between 0 to 65,535 inclusive. The char data type is used to store characters. The size is 2 byte.

(4) Short datatype:- This datatype is a 16-bit Signed two's complement integer. Its value ran is from -32,768 to 32,767 (inclusive) and the size is 2 byte.

(5) Int Datatype:- The int datatype is a 32-bit Signed two's complement integer. It's value-rang lies between 2,147,483,648 (-2^{31}) to 2,147,483,64 $(2^{31}-1)$ inclusive; The size of int datatype is 4 byte.

(6) Long datatype:- The long datatype is a 64-bit two complement integer. It's value-range lies between -9,223,372,036,854,775,808 (-2^{63}) to 9,223,372,036,854,776,807 $(2^{63}-1)$ (inclusive) The size of Long datatypes is 8 byte.

(7) float datatype:- The float data type is a Single-precision 32-bit IEEE 754 floating points. It value range is unlimited. The float data type should never be used for precise values, such as currency. It's default value is 0.0f Size is 4 byte.

(8) Double Datatype:- The double datatype is a double-precision 64-bit IEEE 754 floating point. It value range is unlimited. The double data type is generally used for decimal values just like float. The double data type also. should never be used for precise values such as currency. Its default value is 0.0d. The size is 8 byte.

Q.6) Explain about memory managements in java with reference to stack and heap.

→

In java, Memory Management in a vital process It is managed by Java automatically. The JVM divides the memory into two parts : stack memory and heap memory. These both memory areas are used for different purposes. The major difference between stack memory and heap memory is that the stack is used to store the order of method execution and local variable while the heap memory stores the objects and it uses dynamic memory allocation and deallocati

Stack:- The stack memory is a physical space (in RAM) allocated to each thread at run time. It is created when a thread creates. Memory Management in the stack follows LIFO (Last in First out) order because it is accessible globally. It stores the variables, references to object and partial results. Memory allocated to stack lives until the function returns. If there is no space for creating the new objects, it throws the java.lang.StackOverFlowError.

Heap Memory:-

- It is created when the JVM start up and used by ~~its~~ the application as long as the application runs. It stores object and JRE classes. Whenever we create objects it occupies space in heap memory while the reference of that object creates in the stack. It does not follow any order like the stack. It dynamically handles the memory blocks. For managing the memory automatically java provides the garbage collector that deletes the objects which are no longer being used. The elements are globally accessible in the application. It is a common memory space shared with all the threads.

Q.8) Explain about the terms : Narrowing, widening.

→

Widening Casting :

i] Converting a lower datatype to a higher datatype is known as widening. This case the casting /conversion is done automatically therefore, it is known as implicit type casting. In this case both data types should be compatible with each other.

byte → short → int → long → float → double

Ex:-

```
class Widening Example
{
  public static void main (String args[])
  {
    char ch='c';
    int i=ch;
    System.out.println(i);
  }
}
```

>> Integer value of the given character : 67

Narrowing :-

- Converting a higher data type into a lower one is called narrowing type casting. It is also know as explicit conversion or casting up. It is done manually. by the programmer. if we do not perform casting then compiler reports a compile-time Error

- Narrowing a wider/bigger primitive type value to smaller primitive value type.
- Narrowing a superclass reference to a subclass reference during inheritance.
- We have also performed the narrowing type casting two times. first, we have converted the double type into long data type after that long data type is converted into int data type.

Q.8) Write a short note on static keyword.
→
- The static keyword in java is mainly used for memory management. The static keyword in java is used to share the same variable or method of a given class:-
- The users can apply static keywords with variables, methods, Blocks and nested classes.
- The static keyword belongs to the class that an instance of a class.
- The static keyword is used for a constant variable or a method that is the same ofor every instance of a class.

The static keyword is non-access modifier in java that is applicable for the following:-
1. Blocks
2. Variables
3. Methods.
4. classes.

* Characteristics of static keyword.

i) Shared Memory allocation : Static variables and methods are allocated memory space only once during the execution of the program.

ii) Accessible without object instantiation: Static members can be accessed ; without the need to create an instance of the class.

iii) Associated with class, not objects : Static members are associated with the class, not with individual object.

iv) Cannot access non-static members :-
Static methods and variables cannot access non-static members of a class, as they are not associated with any particular instance of the class.

v) Can be Overloaded, but not overridden:
Static methods can be overloaded, which means that you can define multiple methods with the same name but different parameter.

- static block:-

Ex:-
```
import java.
Public Class block{
  Static int i=10;n;
   static {
    system.out.println ("static block")
     n=i*8;
   }
  Public static void main (string ~~[]~~ args [])
  }
  System.out.println("inside main");
  System.out.println ("value of i" +i);
  System.out.println ("value of n"+n);
 }
}
```

```
>> static block.
   inside main
   value of i=10.
   value of n:80.
```

Q.9) Write a short note on access Specifien.

→

- Access Specifiers are the keywords like "public", "protected", "default" and "Private" which has its Special meaning in Java

a] Public access Specifiers.

- Public is a keyword which is introduced in java
- The access scope of the "Public" is everywhere like in all classes and method as well.
- If we prefixed "Public" keyword with any class, variable or method that it can be accessed by any class or method.

b] Protected access Specifiers.

- Protected is the keyword which is introduced in java.
- The accen Scope of "Protected" is not everywhere and it is accesible in the Same class or its child class or in all those classes which are defined in the Same package.
- If we prefixed "Protected" keyword with any class, variable or method then it can be accesied by the same class or its child classes or all the classes which are defined in the Same packages

c] default access Specifiers

- The access scope of the "default" is not everywhere.
- It is not mandate to prefixed "default" keyword with any class, variable or method because by default class, variable or method is default public in java & it can be accessed by all those classes. which are defined in Some packages only.

d] Private access Specifiers.

- The access scope of the "Private" is not everywhere.
- If we prefixed "Private" keyword with any variable or method then it can be accessed only in the same class.