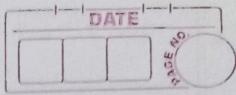


ASSIGNMENT - 1



1. Write a short note on Java program development kit?
- A i] A Java development kit (JDK) is a program development environment for writing Java application. It includes the Java runtime environment (JRE), an interpreter / loader (JAVA), a compiler (JAVA), an archiver (JAR), a documentation generator (javadoc) and other tools needed in Java development.
- ii] You need the JDK to convert source code into a format that the Java runtime environment (JRE) can execute.
- iii] The Java runtime environment itself virtual machine (JVM), supporting files & core classes.
- iv] If you are only interested in running Java programs on your machine or browser, install.
- v] If you would like to develop an application and do Java programming you need JDK.

2. List and explain the salient features of Java.
- A i] Simple : Its syntax is simple, clean & easy to understand.
- ii] Object-oriented : Everything in Java is an object.
- iii] Portable : Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.
- iv] Platform Independent : It is different from other language like C, C++.
- v] Secured : We can develop virus free system.
- No explicit pointer
 - Class loader
 - bytecode verifier
 - Security manager

- 5] Robust - It is robust because it uses strong memory management - there is lack of pointers that avoid security problems.
- 6] ~~Object~~ Architecture Neutral - It is architecture neutral, because there are no implementation dependent features, for eg, size of primitive type is fixed.
- 7] portable - It is portable because it facilitates you to carry java bytecode to any platform it doesn't require any implementation.
- 8] High performance - Java is faster than other traditional interpreted programming language because Java bytecode ("class") is native code - it is still bit slower than a compiled languages, Eg: C++.
- 9] Distributed - It is because it facilitates user to create distributed applications in Java. RMI and ~~EJB~~ JMS are used for creating distributed applications.
- 10] Multi-threaded - A thread is like a separate program; executing concurrently can limit java programs that deal with many tasks at once by defining multiple threads.
- 11] Dynamics - It supports the dynamic loading of classes it means choose, are loaded on demand.

Q.3 Write in detail about different types of operator in java category wise quoting these functional operands and return type, give example.

A i) Arithmetic operator : It is useful for executing addition, multiplication, division, subtraction and modulus.

Eg : public class A {
 public static void main (String [] args)
 { int a = 10 ;
 int b = 20 ;
 System.out.println (a+b);
 System.out.println (a-b);
 System.out.println (a * b);
 System.out.println (a / b);
 System.out.println (a % b); } }

Output :-

- 10

200

0.5

2

ii) Unary Operators :- Unary operators need only one operand these are used to increment, decrement and negative value.

i) - : unary minus used for negating the values.

ii) + : unary plus indicates the positive value if performs automatic conversion to int when the type of operand is byte, character short.

iii) ++ : increment operator used for increment the value by 1. there are two types

post-increment : value used for computing the result then increment

pre-increment - value is incremented first then the result is computed.

iv] -- : decrement operator is used for decrement value by 1. Two types of operation 2 types.

Post-decrement = value is first used for computing result and then decrement

Pre-decrement = value is decremented and then computing results

v] ! = logical not operator used for inverting a boolean value

Eg :- import java. i.e. * ; , || C
class GFG { . . . }

```
public static void main (String [] args)  
{ int a = 10 ;
```

```
    System.out.println ("Post increment :" + (a++));  
    System.out.println ("Pre increment :" + (++a));
```

```
    System.out.println ("Post decrement :" + (a--));  
    System.out.println ("Pre decrement :" + (--a));
```

```
. . .  
}
```

Output :-

post increment = 10 ..

pre increment = 12 ..

post decrement = 10 ..

pre decrement = 8 ..

3] Assignment Operator : Assignment operator is used to assign a value to any variable it has right to left associativity i.e. value given on the right hand side of operator is assigned to variable on the left and therefore right-hand side value must be declared before

i) $=$:- for adding left operand with right operand and then assigning it to the variable on the left

ii) $-=$:- for subtracting the right operand from left operand and then assigning it to the variable on the left

iii) $*=$:- for multiplying left operand with the right operand and then assigning it to the variable on the left

iv) $/=$:- for dividing left operand with the right operand and then assigning it to variable on left

v) $\% =$:- for assigning the remainder of the left operand by right operand and then assigning to variable on the left

Eg:- import java.util.*;
class GFG {

 public static void main (String [] args)
 { int a = 7;

 System.out.println ("a + = 3": + (a += 3));

 System.out.println ("a - = 2": + (a -= 2));

 System.out.println ("a * = 4": + (a *= 4));

 System.out.println ("a / = 3": + (a /= 3));

 System.out.println ("a % = 2": + (a %= 2));

Output :

$a + = 3 : 10$

$a - = 2 : 5$

$a * = 4 : 28$

$a / = 3 : 2.4$

$a \% = 2 : 0.14$

4) Bitwise Operator : These operation are used to perform the manipulation of individual bits of numbers they can be used with any of integer types.

- i) & = returns bit by bit AND of input values.
- ii) | = returns bit by bit OR of input values
- iii) ^ = returns bit by bit XOR input values
- iv) ~ = unary operator which returns the one's complement representation to input values

Eg : import java.util.*;
class GFG {

```
public static void main (String [] args)
{ int d = 011010, e = 0b1100
```

```
System.out.println ("d & e :" + (d & e));
```

```
System.out.println ("d | e :" + (d | e));
```

```
System.out.println ("d ^ e :" + (d ^ e));
```

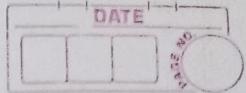
```
System.out.println ("~d :" + (~d));
```

33

5) Shift Operator : It is used to shift all the bits value to the particular side of a specified number of times.

Eg : Public class A {

```
public static void main (String args[])
{
```



System.out.println(10<<2);
System.out.println(10<<3);
System.out.println(20>>2);
System.out.println(20>>3); 33
Output:
40
80
5
2

Q] Relational Operator :- Relational operator are used to check relationship between two operands:

- i) == : checks if two values are equal.
- ii) != : checks if two values are not equal.
- iii) > : Check left operand is greater than right.
- iv) < : check right operand is greater than left.
- v) >= : check if left operand is greater than or equal to right operand.
- vi) <= : check if left operand is less than or equal to right operand.

Eg:- import java.io.*;
public class A {
 public static void main (String [] args)
 {
 int a=5, b=7;
 System.out.println ("a==b", +(a==b));
 System.out.println ("a!=b:", +(a!=b));
 System.out.println ("a>=b:" +(a>=b));
 System.out.println ("a<=b:" +(a<=b));
 System.out.println ("a>b:" +(a>b));
 }
}

Output :

$a == b$: false

$a != b$: true

$a > b$: false

$a < b$: true

$a \geq b$: false

$a \leq b$: true

$a >= b$: true

Q4. What are the primitive data types in Java.

Explain their size, range, and other details.

→ A Primitive data types are the building blocks of data manipulation. There are the most basic data type.

↳ Primitive data type :-

i) Boolean :- It is used to store only 2 possible values, true and false.

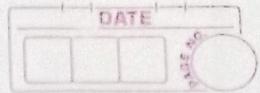
Eg :- Boolean are false.

ii) byte :- It is an 8-bit signed two's complement integer. Its value range lies between -128 to 127. Default value is 0.

Eg :- byte a = 10 ; byte b = -20.

iii) short :- It is a 16-bit signed two's complement integer. Its value range lies between -32,768 to 32,767. Its default value is 0.

Eg :- short s = 1000 ; short r = 5000



4) Integer : The int data type is 32 bit signed two's complement integer its value range lies between $-2,147,483,648$ to $2,147,483,647$, its default value is 0.

Eg : `int a = 1000000`, `int b = 2000000`

5) long : It is 64-bit two's complement integer its value range lies between $-9,223,372,036,854,775,888$ to $9,223,372,036,854$

Eg : `long a = 1000001`, `long b = 20000001`.

i) floating : its value range is unlimited. It is recommended to use a float if you need to save memory in large array of floating no's.

Eg : `float f9 = 234.5F`

ii) double : It is 64-bit IEEE 754 floating point its value range is unlimited, the double data type is generally used for decimal value just like float.

Eg : `double d1 = 12.3`

iii) Char : It is single 16 bit unicode character its value range lies between '1000 + UFFF'

Eg : `Char letterA = 'A'`

5] Explain about memory management in Java with reference.

- A Memory management in Java refers to the process of allocating and freeing up space for objects. Java automatically manages memory. The "garbage collector" is an autonomous memory management technique used in Java.
- i] Heap memory is used by all parts of the application whereas stack memory is used only by one thread of execution.
- ii] Whenever an object is created, it's always stored in the heap space and stack memory contains the reference to it.
- iii] Stack memory only contains local primitive variable to objects in heap space.
- iv] Objects stored in the heap are globally accessible whereas stack memory can't be accessed by other threads.
- v] Memory management in stack is done in LIFO manner whereas it's more in LIFO manner whereas it's more complex in heap memory because it's used globally. Heap memory is divided into young generation.
- vi] Stack memory is short-lived whereas heap memory lives from the start till the end of application execution.
- vii] Stack memory size is very less when compared to heap memory. Because of simplicity in memory allocation (LIFO), stack memory is very fast when compared to heap memory.



6] Explain the terms, narrowing, widening.

A) Widening Casting

i] Widening also known as upcasting is a conversion that implicitly takes ~~with~~.

widening takes place when a smaller primitive type value is automatically accommodated in a larger primitive data type.

Widening also takes place when a reference variable of a superclass is automatically accommodated in reference variable of its subclass.

2] for example - The conversion between numeric data type to char or boolean is not done automatically.

Narrowing Casting.

i]. Converting a higher data type into a lower one is called narrowing type casting. It is also known as explicit conversion or casting up. It is done manually by the programme. If we do not perform casting then compiler reports a compile time error.

double → float → long → int → char → short → byte

2] Narrowing a wider / bigger primitive type value to smaller primitive type values.

3] Narrowing a superior reference to a subclass reference during inheritance.

4] 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 types.

7. Write in detail about static keyword
- i) The static keyword in java is mainly used for memory management. The static keyword is, java is used to share the same variable or method of o gives class.
 - ii) The user can apply static keywords with variable methods, Blocks and nested classes.
 - iii) The static keyword belongs to the class than an instance of a class.
 - iv) The static keyword is used for a constant variable or a method that is the same for every instance of a class.

The static keyword is a non-class access modifier in java that is applicable for the following:

- i) Blocks.
- ii) Variables
- iii) Methods
- iv) Classes.

Characteristics of Static Keyword:

- i] Shared memory allocation : Static variables and methods are allocated memory space only one 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 class.



Q8 Write a short note on access specifier in java

A Access specifiers are the keywords like "Public", "Protected", "default" and "Private" which has its special meaning in Java.

i] Public access specifier

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

ii] Protected access specifier

- "Protected" is the keyword which is introduced in java
- The access scope of the protected is not everywhere and it is accessible in the same class or its child class or in all those classes which are defined in the same package.

iii] Default Access Specifier

- The access scope of the "default" is not everywhere
- It is not mandatory to prefix "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.

iv] Private Access Specifier

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

* * * * *

Q.9 List and explain the components of JVM.

A i] Class loader :

Class loader is used to loads the class file into memory. Whenever we executed the java program. Class loader loads it first. There are three built in class loaders are available in Java :

i) Bootstrap Class loader

ii) Extension Class loader

iii) Application Class loader

ii] Method Area :

Method area stores the all the class level data such as runtime constant pool, method data and code for methods. There is only one method area.

iii] Heap :

Heap memory created when JVM start up. It stores all the objects created during program execution and their corresponding instant variable. It is the run-time data area from which memory for all class instances and array is allocated. There is only one heap area.

iv] Stack :

Java stack memory stores frames, local variables, method calls, partial results.

Whenever a new ~~method~~ thread is created in the JVM, a separate JVM is also created at the same time for that thread.

vii) Program Counter (PC) Registers:

Program Counter register stores address of the currently executing JVM instruction. JVM supports multiple threads, so each thread has its own PC register.

~~Virtual Machine, Compiler~~

viii) Native libraries :-

Java native libraries provide a collection of libraries a collection of libraries which are written in other programming languages needed by execution engine.