4/2/04 ASSIGNMENT-1

PAGE NO.

1] Write a short note on java Development kit.

→ i) A java Development kit (JDK) is a program development environment for writing java applets and applications. It includes the java runtime environment (JRE), an interpreter/ loader (Java), a compiler (java c), 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, and core classes.

iv) If you are only interested in running java programs on your machine or Browser instal JRE.

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.

→ 1] Simple :- Its syntax is simple, clean & easy to understand.

2] object-oriented :- Everything in java is an object.

3] portable - java is portable because it facilitates you to carry the java bytecode to any platform. it doesn't require any implementation.

4] platform Independent - It is different from other languages like C, C++.

5] secured :- you can develop virus-free systems
- No explicit pointer
- class loader

· bytecode verifier
· Security manager
6] Robust :- It is Robust because it uses strong memory management there is lack of pointer that avoids security problems
7] Architecture Neutral :- It is architecture natural because there are no implementation dependent features, for e.g: size of primitive type is fixed.
8] Portable :- It is portable because it facilitates you to carry java bytecode to any platform it doesn't require any implementation.
9] High performance :- Java is faster than other traditional interpreted programming language because java bytecode "close" to native code it is still bit slower than a compiled languages eg:- C++
10] Distributed :- It is because it facilitated users to create distributed applications in java. RMI and EJB are used for creating distributed applications.
11] Multi- threaded :- A thread is like a seperate program, Executing conarntly can write java programs that deal with many tasks at once by defining multiple threads
12] Dynamic :- It supports the dynamic loading of classes. it means choose, are loaded on demand.

3] Write in detail about different types of operator in java category wise quoting their functional operands and return type give are example statement for each.

→ ① Arithmetic operator :- It is useful for executing addition, multiplication, division, subtraction and modules.

e.g :-
```
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 :-
```
30
-10
200
0.5
2
```

② Unary operators :- unary operators need only are operand there are used to increment, decrement or negate a value.

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

ii] + : Unary = phis indicates the positive value it performs automatic conversion to int when the type of operated is byte character short.

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

• post-increment value is first 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 operators two types.

• post-decrement = value is first used for computing result and then decremented
• pre-decrement = value is decremented and then computing result.

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

eg :-
```
import java.io.*; //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 two 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 left.

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

iv] /= :- for dividing left operand with the ~~right~~ right operand and then assigning it to variable on left.

v] %= :- for assigning the module of the left operand by right operand and then assigning to variable on left.

e.g = import java.ia *;
```
    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] & : return bit by AND of input values
ii] | : return bit by OR of input values
iii] ^ : return bit by bit XOR input values
iv] ~ : Unary operator which return the one's complement representation to input value.

eg:-
```
import java.ia *;
  class GFG {
  public static void main (string args [])
  { int d= 011010 , e =0101100.
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));
     } }
```

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

```
e.g =  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);   }}
o/p :- 40
       80
       5
       2
```

⑥ Relational operator = Relational operator are used to check relationship between two operands.
i] == : checks if two values are equal.
ii] != : check 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 righ 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));
system.out.println (" a<b: " + (a<b));
```

output:-
```
a==b : false
a!=b : true
a> =b : false
a<= b : true
a >b : false
a<b : true.
```

4] What are the primitive data types in java? explain their sine, range and other details.

→ Primitive data types are the building blocks of data manipulation there are the most basic datatype.

① Boolean :- It is used to store only two possible values, true and false.
eg:- Boolean are = false.

② byte :- It is an 8-bit signed two's complement integer its value range lies between 128 to 127 its default value is 0.
eg - byte a=10, byte b = -20.

③ Short :- It is a 16-bit signed two's complement integer it value-range lies between -32,768 to 32,767 its default value is 0.
eg :- Short s=10000 , short r=-5000

REDMI NOTE 10 LITE
KAVYA..😇

PAGE NO.:

④ integer :- The int data type is 32-bit signed two's complement integer its value range lies between 2,157,483,648 to 2,147,483, its default value is 0.
e.g in a = 100000 , int b = -2000000.

⑤ long :- It is 64-bit two's complement integer its value - range lies between -9,223,372,036, 854,775,808 to 9,223,372,036,854.
eg :- long a = 1000001 , long b = -20000001.

⑥ Floating :- Its value range is unlimited, it is recommanded to use a float if you need to save memory in large array of floating no's.
eg :- float f1 = 234.5F.

⑦ 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.

⑧ char := It is single 16-bit unicode characters its value range lies between '1000 to UFFF'
eg :- char letter A = 'A'

REDMI NOTE 10 LITE
KAVYA..

5] Explain about memory management in java with reference to stack and heap.

→ 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.

1] Heap memory is used by all the parts of the application whereas stack memory is used only by one thread of execution.

2] Whenever an object is created, it's always stored in the heap space and stack memory contains the reference to it.

3] Stack memory only contains local primitive variable to objects in heap space.

4] Objects stored in the heap are globally accessible whereas stack memory can't be accessed by other threads.

5] 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, old generation etc, more

6] Stack memory is short-lived whereas heap memory lives from the start till the end of application execution.

7] 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.

REDMI NOTE 10 LITE
KAVYA..

6] Explain the terms : narrowing, widening.
→ widening casting.

1] widening also known as upcasting, as a conversion that implicity takes
• widening takes place when a smaller primitive type value is automatically accommodated in a large/wider primitive data type.
• Widening also takes place when a reference variable of a Subclass is automatically accommadated in reference variable of its superclass.

2] For example :- The conversion between numeric data type of char or Boolean is not done automatically.

Narrowing casting.
1] 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 programmer. 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 value.
3] Narrowing a supperclass 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 type.

REDMI NOTE 10 LITE
KAVYA..😇

7] Write in detail about static keyword.

→ 1] 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.

2] The users can apply static keywords with variables, methods, Blocks and nested classes

3] The static keyword belongs to the class than an instance of a class.

4] 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 non-access modifier in java that is applicable for the following

1. Blocks
2. variables
3. Methods
4. classes

Characteristic of static keywords

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

ii) A accessibile 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

REDMI NOTE 10 LITE
KAVYA..

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 parameters.

8] Write a short note on access specifiers in java.
→ Access specifiers are the keywords like "Public", "protected", "default" and "Private" which has its special meaning in java.
i] Public access specifiers
• public is a keyword which is introduced in java.
• The access space of the "public" is everywhere like in all classes and methods as well.
• If we prefixed "public" keyword with any class, variable or method then it can be accessed by any class or methods.
ii] Protected access specifiers
• "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.
• If we prefixed "protected" keyword with any class, variable or method then it can be accessed by the same class or its child classes or all the classes which are defined in the same package.

iii] Default access specifiers
• The access scope of the "default" is not everywhere.
• It is not mandated to prefixed "default" keyword with any class variable or method because by default class, variable or method is default public in java and it can be accessed by all those classes which are defined in same package only
iv] 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.

9] List and explain the components of JVM.
→ 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 built in class loader are available in Java:
i) Boatstrap class loader.
ii) Extension class loader
iii) Application class loader.

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

REDMI NOTE 10 LITE
KAVYA..😇

3> Heap:
Heap memory created when JVM starts. up it stores all the objects created during program execution and their comapand my instant variable. It is the run time data area from which memory for all instances and array is allocated. There is only one heap area.

4> Stack:
Java stack memory stores frames locate variables, method calls, partial results. Whenever a new thread is created in the JVM, a separate JVM is also created at the same time for that thread.

5> Program counter (PC) Registors:
Program counter register stores address of the currently executing JVM construction. JVM support Multiple threads, so each thread has its own PC register.

6> Native Libraries:
Java native libraries provide a collection of Libraries a which are written in other pegrancy languages needed by execution engine.