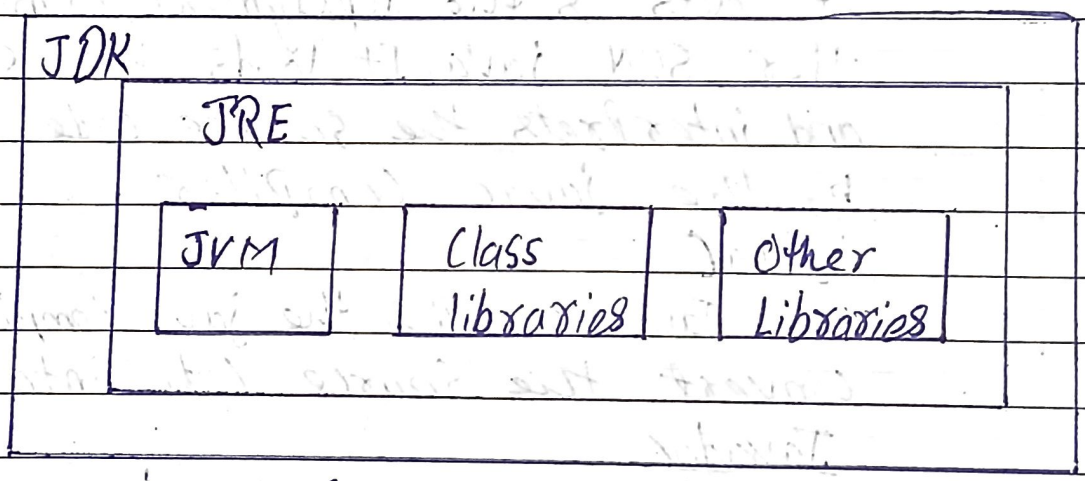
Date
Page

Assignment-1

Q1) Write a short note on Java Development Kit

Ans) Java Development Kit is a bundle of software development tools and supporting libraries combined with the Java Runtime Environment (JRE) and Java Virtual Machine (JVM)



Java Virtual Machine

JVM is a software tool responsible for creating a run-time environment for the Java source code to run. The very powerful feature of Java, "Write Once and run anywhere", is made possible by JVM. The JVM stays right on top of the host operating system and converts the java source code into Bytecode (Machine language), and executes the Program Java Run-time Environment.

JRF is a Software Platform where all the Java Source codes are executed. JRF is responsible for integrating the Software Plugins, jar files, and support libraries necessary for the source code to run.

Components of JDK in Java

Java

It acts is the deployment launcher in the older SUN java. It loads the class files and interprets the source code compiled by the javac compiler.

JavaC

The Javac specifies the java compiler to convert the source code into bytecode.

Javadoc

The javadoc generates documentation for the comments added in the source code.

Jar

The jar helps the archives to manage the jar files in the Package library.

Q2) List and explain the salient features of Java.

Ans2) ① Simple

Java is designed to be easy to learn. Its Syntax is quite simple, clean and easy to understand. Java is inspired by C and C++. There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java

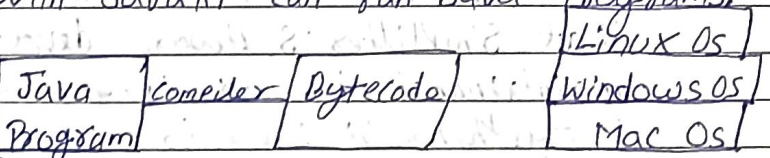
② object oriented

In java, everything is an object which has some data and behaviour. Java can be easily extended as it is based on object model. OOPs is a methodology that simplifies software development and maintainance by providing some rules. Everything in Java is written in terms of classes and objects. Basic Concept of OOPs.

i) Object
ii) Class
iii) Inheritance
iv) Polymorphism
v) Abstraction
vi) Encapsulation

③ Platform Independent

Java is a write once, run anywhere language. A Platform is the hardware or Software environment in which a Program runs - Java Provides a Software -based Platform - Java code can be executed on multiple Platforms, like Windows, Linux, Sun Solaris, Modos etc. On compilation Java Program is compiled into bytecode This bytecode is Platform ~~is~~ independent and can be run on any machine. Plus this bytecode format also Provide Security. Any machine with JavaRF can run Java Programs.



④ Architecture-neutral

Java compiler generate an architecture-neutral object file format, which makes the compiled code executable on many Processors, with the Presence of Java runtime System.

⑤ Portable

Being architecture-neutral and having no implementation dependent aspects of the Specification makes Java Portable Everything related to Storage is Predefined.

⑥ Robust

* Java uses strong memory management

Q3) List and explain the components of Java virtual Machine

Ans Components of Java Virtual Machine:

① Class loader

Class loader is a subsystem of JVM which is used to load class file. Wherever a "java" source file is compiled, it is converted into byte code as a "class" file. When this class is used the class loader loads it into the main memory. The first class to be loaded into memory is usually the class that contains the main() method. There are three Phases.

* Loader

This Phase loads files from Secondary memory into the main memory for execution Loading involves taking the binary representation of a class or interface with a particular name and generating the original class or interface from that

* Linking

Linking is a class or interface involves combining the different elements and the dependencies of Program together. Linking include:

→ Verification:- This Phase checks the structural correctness of the "Class" file by checking. It against a set of constraints or rules.

→ Preparation:- The JVM allocates memory for the static fields of a class or interface, and initiates them with default values.

→ Resolution:- Symbolic references are replaced with direct references in the runtime Constant Pool.

→ Initialization:- Initialization involves executing the initialization method of the class or interface. This can include calling the class's constructor executing the static ~~variables~~ block, and assigning values to all the static variables.

② Runtime Data Area

* Method Area:- Load all the class information used to keep type information about the classes. The method area is created on the virtual machine start-up and there is only one method area Per JVM

* Heap Area:- This is the run-time data area from which memory for all class instances and arrays is allocated. There is only one heap area Per JVM.

* Stack Area :- Whenever a new thread is created in the JVM, a Separate runtime stack is also created at the same time All local variablac method calls, and Partial results are stored in the Stack area.

* Program Counter Registers :- The JVM supports multiple threads at the same time. Each thread has its own Pc Register to hold the address of the Currently executing JVM instraction Once the instruction is executed the PC register is updated with the next instruction

* Native Method Stacks :- The JVM Contains stacks that support native methods. These methods are written in a language other than Java. Such as C and C++. for every new thread, a Separate native method stack is also allocated.

③ Execution Engine :-

once the byte code has been loaded into the main memory, and details are available in the runtime data area, the next step is to run the Program. The Execution Engine handles this by executing the code Present in each class.

→ Interpreter :- The interpreter reads and executes the bytecode instructions line by line.

→ JIT Compiler :- The JIT compiler overcomes the disadvantage of the interpreter. When the interpreter finds same repeated code it uses the JIT compiler.

→ Garbage Collector

collects and removes unreferenced objects from the heap area. It is the Process of reclaiming the runtime unused memory automatically by destroying them.

* Mark :- GC identifies the unused objects.

* Sweep :- removes the objects identified.

④ Native Method Libraries :-

Native Method libraries are libraries that are written in other Programming languages such as C, C++, and assembly. These libraries are usually Present in the form of "dll" or "so" files. These native libraries can be loaded through JNI.

Q4 Write in detail about different type of operators in Java, category wise stating there functionality, operands and return type given one example statement for each.

Ans Arithmatic operators:-

The basic Arithmatic operators - addition, subtraction multiplication and divison - all behave as expected for all numeric types. The unary minus operator negates its single operand. The unary plus operator simply returns the value of its operand.
Operand int, float, double etc. (numeric) - Return type arithmetic

```
class BasicMath {
Public Static Void main (String args[]){
System.out.Println ("Arithmatic Operators");
    int a= 1+1;
    int b= a*3;
    int c= b/4;
    int d= c-a;
    int e= -d;
System.out.Println ("a= "+a);
System.out.Println ("b="+b);
System.out.Println ("c="+c);
System.out.Println ("d ="+d);
System.out.Println ("e="+e);
}
}
```

o/p

```
Arithmetic Operations
a=2
b=6
c=1
d=-1
e=1
```

It also includes

%	Modulus // a%b
++	Increment // a++
--	Decrement // a--
+=	Addition assignment
-=	Subtraction assignment
/=	Division assignment
%=	Modulus assignment
*=	Multiplication assignment

Bitwise Operators

There operators act upon the individual bits of their operands. They are

~	Bitwise unary NOT
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
>>	Shift right
>>>	Shift right zero fill
<<	Shift left

```
System.out.Println("b="+b);
System.out.Println("a|b="+c);
System.out.Println("a&b="+d);
System.out.Println("a^b="+e);
System.out.Println("!a&b|a&!b="+f);
System.out.Println("!a="+g);
    }
}
```

o/p

```
a=true
b=false
a|b=true
a&b=false
a^b=true
!a&b|a&!b=true
!a=false
```

Ternary operator:-

Replace certain types of if-then-else statements

Operands = Boolean expressions.

Return: Based on operands (any)

expression1 ? expression2 : expression3

Ex:-

```
int x=10,y=15;
int result=(x>y)?x:y;
System.out.Println(result);
```

o/p

15

Date
Page

&=	Bitwise AND assignment
|=	Bitwise OR assignment
^=	Bitwise exclusive OR assignment
>>=	Shift right assignment
>>>=	Shift right zero fill assignment
<<=	Shift left assignment.

Operands = Integer type.

```
Class Bitlogic {
Public Static void main (String args[]) {
String binary[] = {"0000","0001","0010","0011"
"0100","0101","0110","0111","1000","1001","1010",
"1011","1100","1101","1110","1111"};
int a= 3;
int b= 6;
int c= a|b;
int d = a&b;
int e = a^b;
int f = (~a&b)|(a&~b);
int g= ~a&0x0f;
System.out.Println(" a= "+binary[a]);
System.out.Println(" b= "+binary[b]);
System.out.Println("a|b= "+binary[c]);
System.out.Println("a&b= "+binary[d]);
System.out.Println("a^b= "+binary[e]);
System.out.Println("~a&b|a&~b= "+binary[f]);
System.out.Println(" ~a= "+binary[g]);
}
}
```

O/P
a = 0010
b = 0110
a|b = 0111
a&b = 0010
a^b = 0101
~a&b|a&~b = 0101
~a = 1100

Relational Operators

Determine the relationship that one operators operand has to the other operands = Any type
Return = Boolean value.

==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Ex:
```
Class Relate{
  Public static void main(String args[]){
    int a=4;
    int b=1;
    boolean c= a<b;
    System.out.Println(c);
  }
}
```

O/P False

Logical Operators:-

Perform logical operations on operands.

Operands = only Boolean type

Return = Boolean.

Op	Result
&	Logical AND
|	Logical OR
^	Logical XOR
||	Short-circuit OR
&&	Short-circuit AND
!	Logical Unary NOT
&=	AND assignment
|=	OR assignment
^=	XOR assignment
==	Equal to
!=	Not equal to
?:	Ternary if-then-else.

Ex:-

```
Class BoolLogic {
    Public static void main(String args[]){
        boolean a = true;
        boolean b = false;
        boolean c = a|b;
        boolean d = a&b;
        boolean e = a^b;
        boolean f = (!a&b) | (a&!b);
        boolean g = !a;
        System.out.println(" a = " + a);
```

Q5 What are the Primitive data types in Java? Briefly explain their size, range and other details.

Ans Primitive Data Types (& types):

① Integers.

This group includes byte, short, int and Long, which are for whole-valued signed numbers. All of these are signed, Positive and negative values.

Long

size = 64 bits

range = -9,223,372,036,854,775,808 to 9,223,372,036,854,775 807

Uses: useful for those occasions where an int type is not large enough to hold the desired value.

Int

size = 32 bits

range = -2,147,483,648, to 2,147,483,647

Uses: When byte and short values are used in an expression, they are Promoted to int when the expression, is evaluated. So, int is the most commonly used integer type. It is the best choise when an integer is needed.

Short
Size = 16 bits
range = -32,768 to 32,767
Uses = least used Java type larger than byte.

Byte
Size = 8 bits
range = -128 to 127
Uses: Used while working with stream of file or network and with raw binary data.

② Floating-Point Types
- Also known as real numbers are used when evaluating expressions that require fractional Precision.

double
Size = 64 bits
range = 4.9e-324 to 1.8e308
Uses = It is actually faster than Single Precision on some modern Processors that have been optimized for high-speed mathematical calculations.

Float
Size = 32 bits
range = 1.4e-045 to 3.4e+038
Uses = It Specifies the Single Precision value that uses 32 bits of storage. Single Precision is faster on some Processors and takes half as much space as double Precision.

③ Characters

Char

Size = 16 bits

range = 0 to 65,536 (no negative chars)

Uses = Represents a single character. or it stores the characters. Char in Java is designed to hold Unicode characters. it can also be used as an integer type.

④ Booleans

range = true or false

Uses = It can have only one of two possible values, true or false. This is the type returned by all relational operators as in the case of acb = boolean is also the type required by the conditional expressions that govern the control statements such as if and for.

Q6 Explain about memory management in java with reference to stack and heap.

Ans The Process of allocation and de-allocation of objects is called memory management. The JVM divides the memory into two parts stack memory and heap memory.

Stack

Stack memory is a Physical space allocated to each thread at run time. It is created when a thread creates Memory management in the stack follows LIFO order because it is accessible globally. It stores the variables, references to objects, and Partial results.

Features

→ Variables inside the stack exist only as long as the method that created them is running.

→ It is automatically allocated and deallocated when the method finishes execution.

→ Access to this memory is fast when compared to heap memory.

→ If this memory is full, Java throws java.lang.StackOverFlow Error.

→ This memory is threadsafe as each thread operates in its own stack.

Heap

It is created when the JVM starts up and used by the application as long as the application runs. It stores objects and JRE classes. Whenever we create objects it occupies space in the heap memory while the reference of that object creates in the stack. It dynamically handles the memory blocks. It means, we need not to handle the memory manually. Java provides the garbage collector that deletes the objects which are no longer being used.

Features

→ If heap space is full, Java throws java.lang.outOfMemoryError.

→ Access to this memory is comparatively slower than stack memory.

→ This memory, in contrast to stack, isn't automatically deallocated. It needs garbage collector to free up unused objects so as to keep the efficiency of the memory usage.

Q7 Explain the terms narrowing, widening.

Ans Narrowing

Converting a larger type to a Smaller size type.

double → Float → long → int → char → Short → byte

Narrowing must be done manually by placing the type in parentheses in front of the value.

Ex:-
```
class Main {
    Public Static void main(String args[])
{
    double d = 9.78d
    int i = (int) d
    System.out.Println (d);
    System.out.Println (i);
}
}
```

O/P
```
9.78
9
```

→ Also Known as explisit conversion

→ Converting higher data type to lower data type.

Widening:
Converting a smaller type to a larger type size.
byte → short → char → int → long → Float → double.

Ex:
```
Class main {
    Public Static void main (String args[])
    {
        int i = 9;
        double d = i;
        System.out.Println(i);
        System.out.Println(d);
    }
}
```

O/P
9
9.0

→ It is also known as implicit conversion
→ It is done automatically.
→ It is safe because there is no chance to lose data - It takes place when
* Both data types must be compatible with each other.
* The target type must be larger than the source type.

Q8 Write in detail about Static Keyword.

Ans 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 static keyword belongs to the class than an instance of the class. The Static keyword is used for a constant variable or a method that is the same for every instance of a class. It refers to the common Property of all objects. The Static Keyword can be applied to.

Variables

→ When a variable is declared as static, it is known as static variable.
→ The Static variable gets memory only once in the class area at the time of class loading.
→ It makes program memory efficient.
→ Static variables are, essentially, global variables.
→ Static variables are created at class level only.

Method

→ When static Keywood is applied with any method, it is known as Static method.

→ A Static keyword method belongs to the class rather than the object of a class.
→ A Static method can be invoked without the need for creating an instance of a class.
→ A Static method can access Static data member and can change the value of it.
→ The most common example of a Static method is the main() method.
→ They cannot refer to this or super in any way.

Classes

→ A Class can be made static only if it is a nested class.
→ Nested Static class doesn't need a reference of outer class. In this case, a Static class cannot access non-static members of the outer class.

Block

→ It is used to initialize the static data member.
→ It is executed before the main method at the time of class loading

Q9 Write a short note on access specifiers in Java.

Ans Access specifiers help to restrict the scope of the class constructor, variable method or data member. It provides security, accessibility, etc to the user depending upon the access specifier used with the element.

① Default

When no access specified is used for a class, method or data member. It is said to be having the default access specifier by default. The data members classes, or methods that are not declared using any access modifiers i.e having default access specifiers are accessible only within the same package

② Public

The Public access specifier is used using the keyword 'public'.

→ The Public access specifier has the widest scope among all other specifiers

→ Classes, methods, or data members that are declared as Public are accessible from every where in the Program.

There is no restriction on the scope of Public data members.

③ Private

→ It is specified using the keyword 'Private'.

→ The methods or data members declared as Private are accessible only within the class in which they are declared.

→ Any other class of the same Package will not be able to access these members.

→ Top-level classes or interfaces can not be declared as Private

④ Protected

→ It is specified using the keyword 'Protected'

→ The methods or data members declared as Protected are accessible within the same Package or Subclasses in different Packages.

→ It Provides more accessibility than the default modifier