Unit I - Introduction to OOPs & Java

[OOPs - Problems in Procedure Oriented Approach, Features of Object Oriented Programming System, OOPs Concepts(Class/Object-Encapsulation, Abstraction, Inheritance, Polymorphism), History and evolution of Java, Data types, variables, arrays, Operators and control structures.

Disclaimer: These notes are just for your basic understanding. Please refer the prescribed text book.

OOPs-Problems in Procedure Oriented Approach: Procedure-oriented programming is one of the earlier programming paradigms which became quite popular. As the name suggests, it focus on the procedures, the code specific instructions to carry out a specific task. There are various advantages of procedure oriented programming languages. It is good for general purpose programming and we can re-use pieces of code in a program without writing them again, by the virtue of method calling. We can structure a program based on these methods and have a cleaner and more maintainable code when compared to the languages it succeeded (e.g Assembly language). There were however, found some flaws in maintaining code that grew too large and became complex and this very though led to the next generation of programming languages which focused more on data rather than instructions. Some common disadvantages of procedural languages are; Data is exposed to whole program at once, so there is no security of data available. Since the focus is on the instructions, it is rather difficult to relate to real world objects and in transition some real world problems. There is no hierarchy in code. These and many more disadvantages combined, led way to a new imperative programming paradigm, i.e, Object oriented programming(OOP).OOP is great at maintaining and testing large code.

Features of Object Oriented Programming System: The OOP language supports all the features of normal programming languages. In addition it supports some important concepts and terminology which has made it popular among programming methodology. The important features of Object Oriented programming are:

- Inheritance

- Polymorphism
- Data Hiding
- Encapsulation
- Overloading
- Reusability
- Objects
- Classes

Objects: An object is an instance of a class.It stores its stste in fiels/variables and exposes its behaviour through methods

Classes: A class is the blueprint from which individual objects are created. These contain data and functions bundled together under a unit. In other words class is a collection of similar objects. When we define a class it just creates template or Skelton. So no memory is created when class is created. Memory is occupied only by object. In other words classes acts as data types for objects.

Member functions: The functions defined inside the class as above are called member functions.

Data Hiding: This concept is the main heart of an Object oriented programming. The data is hidden inside the class by declaring it as private inside the class. When data or functions are defined as private it can be accessed only by the class in which it is defined. When data or functions are defined as public then it can be accessed anywhere outside the class. Object Oriented programming gives importance to protecting data which in any system. This is done by declaring data as private and making it accessible only to the class in which it is defined. This concept is called data hiding.

Encapsulation: The technical term for combining data and functions together as a bundle is encapsulation. Hiding internal state and requiring all interaction to be performed through an object's method is known as data encapsulation.

Inheritance: Inheritance as the name suggests is the concept of inheriting or deriving properties of an existing class to get new class or classes. In other words we may have common features or characteristics that may be needed by number of classes. So

those features can be placed in a common class called base class/parent class/super class and the other classes which have these characteristics can take the class and define only the new things that they have on their own in their classes. These classes are called derived class. The main advantage of using this concept of inheritance in Object oriented programming is it helps in reducing the code size since the common characteristic is placed separately called as base class and it is just referred in the derived class. This provide the users the important usage of terminology called as reusability

Reusability: This is achieved by the terminology called as inheritance. Reusability is nothing but re- usage of structure without changing the existing one but adding new features or characteristics to it. It is very much needed for any programmers in different situations. Reusability helps in reducing the code size since classes can be just derived from existing one and one need to add only the new features and it helps users to save their time.

Polymorphism and Overloading: Polymorphism as the name suggests is a certain item appearing in different forms or ways. That is making a function or operator to act in different forms depending on the place they are present.

History and evolution of Java: In 1990 Sun Microsystems Inc.(US) has conceived a project to develop software for consumer electronic devices that could be controlled by a remote. This project was called Stealth Project but later its name was changed to Green Project.In January 1991 ,Bill Joy, James Gosling, Mike Sheradin,Parick Naughton and several others met in Aspen to discuss this project and decided to develop a completely system independent language.This language was initially named as OAK but later changed to JAVA. Sun formally announced Java and Hot Java at Sun World conference in 1995. Apart from being system independent language there are other reasons too for the immense popularity Java.

Features of Java: Some of the important features which make Java different from other languages are simple, object-oriented, distributed, robust, secure, system independence, portability, interpreted, high performance, multithreaded, scalability and dynamic.

Simple: The same syntax of C and C++ is maintained in Java and difficult concepts like pointers have been completely eliminated.

Object-oriented: Java is a purely object oriented language which has all the five features of OOPS like classes and objects, encapsulation, abstraction, inheritance and polymorphism.

Distributed: Since Java can handle protocols like TCP/IP and UDP it can be effectively used in networking.

Robust: Since Java has efficient exception handling mechanisms and memory management features we can say java is a robust language.

Secure: Security problems like eavesdropping, tampering, impersonation, and virus threats can be eliminated or minimized by using Java on Internet.

System independence: Java's byte code is not machine dependent. It can run on any machine with any processor and any operating system.

Portability: Because of the system independent nature of Java we can call it as a portable language.

High Performance: The JIT (Just In Time) compiler attached to the JVM enhances the speed of execution .In JVM both interpreter and JIT compiler work together to run the program.

Multithreaded: A thread represents an individual process to execute a group of statements.JVM uses several threads to execute different blocks of code. Creating multiple threads is called "multithreaded".

Scalability: Since Java is compact and platform independent, it can be implemented on a wide range of computers –from embedded devices to mainframe computers. **Dynamic:** Before the development of Java only static text used to be displayed in the browser .In Java the applet programs helps to use dynamically interacting programs on Internet.

Eight Primitive Data types

Integer data types represent integer numbers. It is again subdivided into byte, short , int and long. Float data types are useful to represent numbers with decimal point .It is again classified as float and double. Character data type represents a single character. String data type represents a group of

characters Boolean data type represents any two values-true or false

Variables : A variable is an important concept in Java language. It basically causes a compiler to set aside some memory at the time of compilation. A variable is the name given to a memory location. It is the basic unit of storage in a program. The value stored in a variable can be changed during program execution. A variable is only a name given to a memory location, all the operations done on the variable effects that memory location. In Java, all the variables must be declared before use.

There are three types of variables in Java; Local Variables , Instance Variables ,Static Variable. A variable defined within a block or method or constructor is called local variable. These variable are created when the block in entered or the function is called and destroyed after exiting from the block or when the call returns from the function. The scope of these variables exists only within the block in which the variable is declared. i.e. we can access these variable only within that block. Initialization of Local Variable is Mandatory. Instance variables are non-static variables and are declared in a class outside any method, constructor or block. As instance variables are declared in a class, these variables are created when an object of the class is created and destroyed when the object is destroyed. Unlike local variables, we may use access specifiers for instance variables. If we do not specify any access specifier then the default access specifier will be used. Initialization of Instance Variable is not Mandatory. Its default value is 0 . Instance Variable can be accessed only by creating objects. Static variables are also known as Class variables. These variables are declared similarly as instance variables, the difference is that static variables are declared using the static keyword within a class outside any method constructor or block. Unlike instance variables, we can only have one copy of a static variable per class irrespective of how many objects we create. Static variables are created at start of program execution and destroyed automatically when execution ends. Initialization of Static Variable is not Mandatory. Its default value is 0 . If we access the static variable like Instance variable (through object), compiler will

show the warning message and it won't hault the program. Compiler will replace the object name to class name automatically. If we access the static variable without classname, Compiler will automatically append the class name. To access static variables, we need not to create any object of that class, we can simply access the variable as:
classname. variablename

Arrays: An array represents a group of elements of the same data type. In Java arrays are created on dynamic memory i.e. , allotted at runtime by JVM .Arrays are generally categorized into Single dimensional array(One Dimensional array) and multi-dimensional arrays(2D , 3D , ...)Single dimensional array represent a row or a column of elements .The array index starts from 0. There are many ways of creating a single dimensional array .One example of creating a 1D array of integer type and 2D array type String is given below.

```
int marks[ ]=new int[5];
String marks[ ][ ]=new String[10][5];
//Example program class Sum {
public static void main(String[] args)
{
int[] ia = new int[101];
for (int i = 0; i < ia.length; i++)
ia[i] = i; int sum = 0;
for (int i = 0; i < ia.length; i++)
sum += ia[i];
System.out.println (sum); } }
```

Operators: An operator is a symbol that performs an operation .An operator acts on some variables called operands to get the desired result. If an operator acts on a single variable it is called unary operator; if it acts on two variables it is called binary operator and if it acts on three variables then it is called ternary operator. There are different types of operators like arithmetic operators, relational operators, logical

operators, Boolean operators, bitwise operators , ternary operator or conditional operator , member operator, instance of operator , new operator and cast operator.

Control structures: The different control structures in java is the same as that in C and C++.

- if....else statement
- do... while loop
- while loop
- for loop
- switch statement
- break statement
- continue statement
- return statement

Access Specifiers:Java Access Specifiers (also known as Visibility Specifiers) regulate access to classes, fields and methods in Java.These Specifiers determine whether a field or method in a class, can be used or invoked by another method in another class or sub-class. Access Specifiers can be used to restrict access. There are four type of the access specifiers in Java.

- public: accessible in all class in your application.
- protected: accessible within the class in which it is defined and in its subclass(es)
- private: accessible only within the class in which it is defined.
- default (declared/defined without using any modifier) : accessible within same class and package within which its class is defined.