**JAVA NOTES [ANUDIP]**

**Day 1 notes**

**Task:**

**1. Create a repository here for FWJP**

**2. Create a folder for each exercise and upload all your programs.**

**3. To create a folder in the repository**

**-> click on create a new file**

**Java Development process started in 1991. Initially java was developed for set top box technology.**

**which should be platform independent, secure etc.**

**=> Platform independent means a software which can run on any platform(os) without any changes.**

**JVM : Java Virtual Machine : It is responsible for running all the java programs . it provides a layer which interacts with Operating system for running the programs.**

**JRE : Java Runtime Environment : it is responsible for running, loading and unloading of java classes, so that JVM can run those class files.**

**BYTE CODE : it is a special code which we get after the file is compiled(interpreted). This code runs on jvm by jre to execute all the instructions.**

**- core java developer**

**- web developer(adv java, hibernate, spring, spring boot)**

**- mobile app developer**

**class ClassName**

**{**

**declare variables**

**declare constructors**

**declare blocks**

**public static void main(String args[])**

**{**

**instructions**

**}**

**}**

**=> class :**

**-> class is a blueprint of an object.**

**-> class contains objects, variables, methods etc.**

**-> it defines the property and behaviour of the object.**

**=> public : it is an access modifier. It defines the scope of any class, variable, method, constructors etc., which means the access/ limitation of a particular entity.**

**=> static : static is used to defined class level variables, methods, or blocks. these static entities are defined only in the memory of the class, They are not part of any object.**

**=> void : it is a return type, which defines that the method will not return any value to the caller.**

**=> main() : It is the entry point of the program/ project. Jvm accesses the main method whenever we start running our project/ program**

**=> String : String is a pre-defined class of java, which is used for creating string variables, and it is used to store string(Sequence of characters) values. String non-primitive datatype.**

**=> args[] : it is a One-Dimension array of string type. it is used to hold command line arguments.**

**CODE EXAMPLE :-**

**class FirstDemo**

**{**

**public static void main(String args[])**

**{**

**// single line comments**

**/\***

**multiline comment**

**\*/**

**// to print a message on console screen**

**System.out.println("Hello World");**

**}**

**}**

**Day 2 Notes**

**class :**

**-> class is a blueprint of an object.**

**-> class contains objects, variables, methods etc.**

**-> it defines the property and behaviour of the object.**

**Object : instance of a class. object is real world entity.**

**object can be any real world entity which has certain attributes and behaviour. In OOPS, an object is and instance variable of a class, which obtains the access to its variables and methods.**

**An object has a life cycle, they are being used for a particular time frame. After their work gets completed, they get deleted/removed from the memory.**

**Methods : A block of code which is used to perform a particular task. A method can be called multiple times whenever they are needed.**

**Functions are independent entity which can be defined without the class and can be called directly But Methods are instance/member of a class, so it's access is dependent on a class behaviour. they can be called using an object.**

**access-modifier return-type methodname(dtype varname, dtype varname,......)**

**{**

**code**

**statements**

**}**

**To access a method, we need to create an object**

**classname obname = new classname();**

**obname.methodname();-> it will call the method.**

**new -> it is responsible allocating memory for an object, creating instance variables, methods in object memory.**

**classname() -> it is a call to a constructor.**

**Constructor : it is used to initialize class/instance variables. constructor name is same as class. By default every class in java has a default constructor, which is usually hidden, but we can override this constructor to perform more tasks.**

**Access-modifiers**

**1. public -> it can be accessed from any where**

**2. private -> it can be accessed from with in class only**

**3. protected -> it can be accessed from within package but in different package, only child class can access it.**

**4. No modifier -> it can be accessed only within the package.**

**To define a variable-> access-modifier datatype varname;**

**Day 3 :- JAVA**

**Class VarDemo**

**{**

**Public static void main(String args [])**

**{**

**int a =10;**

**int b = 20;**

**int sum;**

**system.out.println(a);**

**system.out.println(“a : “ +a); // a : 10**

**system.out.println(“a : “+a”\tb : “ +b);**

**sum=a+b;**

**system.out.println(sum : “ sum);**

**//+ - \* / %**

**System.out.println(“Sum: + ”(a+b));**

**System.out.println(“difference : + ”(a-b));**

**System.out.println(“Product: + ”(a\*b));**

**System.out.println(“Modulas: + ”(a%b));**

**}**

**}**