**20-09-2021**

**Version Control System : Version control system that records changes on files or program or application in a projects.**

**3 types**

**Local Version control : RCS : Revision control system**

**Centralized version control : SVN**

**Distributed Version control system : Local repository and remote repository.**

**Git : Git is distributed Sub Version control system which help to manage the source code.**

**Distributed version control system keeps track of software version and allow many developers to work on given project within a maintaining connection of common network.**

**Git is a open source distributed version control system.**

**Create one folder My Repository**

**Please create simple text file or any type of time with some contents.**

**Open terminal in Virtual machine.**

**git --version**

**to make normal folder as a local git repository we have to**

**run the command as**

**git init : after run this command it will create .git folder.**

**This folder consider as a hidden folder in Unix or mac**

**git status :This command is use to check the last command status.**

**git add filename This command is use to add the file from local file system to staging area.**

**Staging area : it is a area generally represent in git directory, that store information about what will go to next commit.**

**It is a intermediate area between os file system and git local repository.**

**git commit –m “First commit”**

**all commands**

**git init**

**git status**

**git add filename**

**or**

**git add . : This command is use to add all files and folder**

**git status**

**git commit –m “message”**

**first this command**

**git config –-global user.email “akash300383@gmail.com”**

**second this command**

**git config –-global user.name “Akash”**

**No do the change in same file**

**Remote repository**

**Github is type of remote repository which help to share the data from one developer to another developer.**

**AWS : code commit**

**Gitlab**

**Azure :**

**So first create the account in github**

**Then create the repository.**

**To link local repository to remote repository we have to run the command as**

**Git remote add origin URL**

**git remote add origin** [**https://github.com/Kaleakash/Java\_FIS\_2021\_Batch.git**](https://github.com/Kaleakash/Java_FIS_2021_Batch.git)

**after connect local to remote**

**we have push local repository code to remote repository**

**before push we have to check default branch**

**git branch**

**master/main**

**git push –u origin master**

**or**

**git push –u origin main**

**git branch :**

**git branch is use to hold more than one commit details.**

**Branch is like a pointer which hold more than one commit.**

**Default branch name is master/main**

**git branch (master/main)**

**to create the user-defined branch**

**git branch branchName**

**command to switch to user-defined branch**

**git checkout branchName**

**To delete the branch**

**git branch –D branchName**

**merge the code**

**git merge branchName : This command is use to merge to code to current branch**

**create folder**

**then create the file**

**then git init**

**then git add .**

**git commit –m “message”**

**Syntax**

git remote add origin https://token@github.com/Kaleakash/test\_info.git

**To Generate token**

****

****

****

****

****

****

git remote add origin <https://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@github.com/Kaleakash/test_info.git>

after added url to origin variable we can push the data to remote repository

git push –u origin master

git remote remove origin

Java Training

**This command is use to download the new remote repository code in local machine**

**git clone https://github.com/Kaleakash/Java\_FIS\_2021\_Batch.git**

**After clone move inside a repository folder.**

**This command is use to pull new update from remote repository to local repository.**

**git pull**

**Java Training**

**What is Java.**

**Java is a platform independent and pure object oriented programming language.**

**C 1970**

**C++ 1980**

**Java 1990 :**

**Initial name of is Oak. 1991**

**Rename in Nov 1995 to Java.**

**James gosling and Team**

**Java was belong to Sun micro system but not it is a part of Oracle.**

**Version of Java**

**1.0 to 17**

**Java 8 mandatory to develop the application.**

**object : any real world entity.**

**Properties or state 🡪 have-🡪 fields / variables.**

**Person**

**Behaviour -🡪do/does -🡪 function / methods**

**Place**

**Bank**

**Animal**

**Car**

**class : blue print of object or template of object.**

**Class syntax**

**class classname {**

**fields;**

**methods;**

**}**

**In Java class name must be follow Pascal naming rules.**

**If it contains one world first letter must be upper case. If it contains more than one world each world first letter upper case.**

**Demo.java**

class Demo {

    public static void main(String args[]){

        System.out.println("Welcome to Java");

        System.out.print("Welcome");

        System.out.printf("Welcome to Java");

    }

}

**Data types : it is a type of data which tell what type of data it hold.**

**2 types**

1. **Primitive data types : it is used to store only value** 
   1. **byte 1 byte**
   2. **short 2 byte**
   3. **int 4 byte**
   4. **long 8 byte : without decimal**
   5. **float 4 byte**
   6. **double 8 byte :with decimal**
   7. **char 2 byte :any singe character**
   8. **boolean 1 bit : true or false.**

**Primitive data type example**

class Demo {

    public static void main(String args[]){

        int a=10;

        System.out.println(a);

        System.out.println("Value of a is = "+a);

        System.out.printf("Value of a is = %d\n",a);

    }

}

**Type casting :converting from one data type into another data type is known as type casting.**

**2 types**

1. **implicit type casting**
2. **explicit type casting**

**int family**

**------------------------implicit -------------🡪**

**byte short int long**

**🡨-----------------explicit ---------------------**

**Type casting example**

class Demo {

    public static void main(String args[]){

        byte a =10;

        short b =a;     // implicit type casting

        System.out.println(a);

        System.out.println(b);

        short c = 10;

        // (type)variableName;

        byte d = (byte)c;       // explicit type casting

        System.out.println(c);

        System.out.println(d);

    }

}

**Type casting int and float variables**

class Demo {

    public static void main(String args[]){

        int a=10;

        float b=a;      // implicit type casting

        System.out.println(a);

        System.out.println(b);

        //float c = (float)10.10;

        float c = 10.10f;       // explicit type casting

        int d = (int)c;         // explicit type casting

        System.out.println(c);

        System.out.println(d);

    }

}

**Operator :**

**If statement**

**If else**

**Nested if**

**If else if**

**Switch statement**

**Looping : looping is use to execute the statement again and again till conditions becomes false.**

**Initialization : start and end**

**Condition : true**

**Coding**

**Increment or decrement.**

**While loop**

**Do while loop**

**For loop**

**Looping Example**

class Demo {

    public static void main(String args[]){

        // System.out.println("While loop");    // entry loop

        // int i=1,n=10;

        // while(i<=n){

        //     System.out.println(i);

        //     i++;

        // }

        // System.out.println("Do While loop");    // exit loop

        // int i=1,n=10;

        // do{

        //     System.out.println(i);

        //     i++;

        // }while(i<=n);

        System.out.println("for loop");

        for(int i=0,n=10;i<=n;i++){

            System.out.println(i);

        }

    }

}

**In java every decimal number by default consider as double.**

1. **Non primitive data types or reference data types: it is used to store value as well as reference of another data types.** 
   1. **Array**
   2. **Class : pre-defined or user-defined**
   3. **Interface : pre-defined or user-defined**
   4. **Enum**

**array : array is used to store more than one value of same types.**

**syntax**

**datatype arrayName[];**

**for each or enhanced loop**

**for(datatype variableName : arrayName) {**

**}**

**Array with looping (for loop and enhanced loop)**

class Demo {

    public static void main(String args[]){

        int []abc;

        int []num = {10,20,30,40,50,60};

        System.out.println("Value of 0 position in num "+num[0]);

        System.out.println("Value of 1 position in num "+num[1]);

        System.out.println("using for loop");

        for(int i=0;i<6;i++){

            System.out.println(num[i]);

        }

        System.out.println("for enhanced loop");

        for(int n:num){

            System.out.println(n);

        }

    }

}

**Creating the memory size for the array**

**datatype arrayName[]=new datatype[size];**

**GC()**

class Demo {

    public static void main(String args[]){

        int []abc={10,20,30,40,50};

        int []num = new int[10];

        System.out.println("Size of array is "+abc.length);

        System.out.println("Size of array is "+num.length);

        System.out.println(abc[0]);

        System.out.println(num[0]);

        System.out.println(num[1]);

        int temp =100;

        for(int i=0;i<num.length;i++){

            num[i]=temp;

            temp++;

        }

        for(int i=0;i<num.length;i++){

            System.out.println(num[i]);

        }

    }

}

**Taking the value through keyboards.**

1. **Using Scanner class**
2. **Using DataInputStream**
3. **BufferedReader**
4. **Command line arguments.**

**Scanner is a pre-defined class which provide set of methods which help to scan the value through keyboards.**

**Syntax to create the Scanner class object.**

**Scanner obj = new Scanner(System.in);**

**Scanner pre-defined class part of util package.**

**Package is a collection of classes and interfaces.**

**In Java all method follow camel naming rules.**

1. **If method name contains one word it must be in lower case.**
2. **If method name contains more than one world from second word each word first letter upper case.**

**Taking the value through keyboards using Scanner class**

//import java.util.Scanner;

import java.util.\*;

class Demo {

    public static void main(String args[]){

        String msg = "Welcome";

        System.out.println(msg);

        Scanner obj = new Scanner(System.in);

        System.out.println("Enter the id");

        int id = obj.nextInt();

        obj.nextLine();             // it is use to hold enter key

        System.out.println("Enter the name");

        String name= obj.nextLine();

        System.out.println("Enter the salary");

        float salary = obj.nextFloat();

        System.out.println("id is "+id);

        System.out.println("name is "+name);

        System.out.println("Salary is "+salary);

    }

}

**Taking the value through keyboards**

//import java.util.Scanner;

import java.util.\*;

class Demo {

    public static void main(String args[]){

        Scanner obj = new Scanner(System.in);

        System.out.println("how many number do you want to store?");

        int n = obj.nextInt();

        int []num = new int[n];

        System.out.println("Enter the number one by one");

        for(int i=0;i<n;i++){

            num[i]=obj.nextInt();

        }

        System.out.println("The number are");

        for(int i=0;i<n;i++){

            System.out.println(num[i]);

        }

    }

}

**Take array value as id,name,salary and desg**

**If desg is manager we have to give 5000 bonus if desg is developer 3000 else 1500**

//import java.util.Scanner;

import java.util.\*;

class Demo {

    public static void main(String args[]){

        Scanner obj = new Scanner(System.in);

        System.out.println("how many number do you want to store?");

        int n = obj.nextInt();

        int []id=new int[n];

        String []name = new String[n];

        float []salary = new float[n];

        String []desg=new String[n];

        for(int i=0;i<n;i++){

            System.out.println("Enter the id");

            id[i]=obj.nextInt();

            System.out.println("Enter the name");

            name[i]=obj.next();

            System.out.println("Enter the salary");

            salary[i]=obj.nextFloat();

            System.out.println("Enter the Desg");

            desg[i]=obj.next();

        }

        for(int i=0;i<n;i++){

            if(desg[i].equals("Manager")){

                  salary[i]=salary[i]+5000;

            }else if(desg[i].equals("Developer")){

                   salary[i]=salary[i]+3000;

            }else {

                salary[i]=salary[i]+1500;

            }

        }

        System.out.println("All details are");

        for(int i=0;i<n;i++){

            System.out.println("id is "+id[i]);

            System.out.println("name is "+name[i]);

            System.out.println("salary is "+salary[i]);

            System.out.println("desg is "+desg[i]);

        }

    }

}

**objet : any real world entity**

**properties or state**

**Car**

**Behaviour**

**Person**

**Bank**

**Animal**

**Customer**

**Employee**

**class : it is user-defined data type which help to create the**

**memory or object.**

**Syntax to create the object**

**ClassName objectRefName = new ClassName()**

**objectRefName.methodName();**

class Car {

    int wheel;

    float price;

    String color;

    void start() {

        System.out.println("Car Start");

    }

    void appliedGear() {

        System.out.println("Applied Gear");

    }

    void moving() {

        System.out.println("Car is moving");

    }

    void stop() {

        System.out.println("Car Stop");

    }

}

class CarTest {

    public static void main(String args[]){

        //System.out.println("Main method");

        //start();

        Car innova = new Car();     // heap memory

        innova.start();

        innova.stop();

    }

}

**Types of variable or fields**

**In Java variables are divided into three types.**

1. **Instance variable**
   1. **The variable which declare outside a method but inside a class is known as instance variable.**
   2. **Instance variable hold default value according to their data types. like int family 0, float family 0.0, char space, String null, boolean false.**
   3. **We can use instance variable directly inside a method but the method must be non-static and it must be part of same class.**
2. **Local variable**
   1. **The variable which declare inside a method is known as local variable.**
   2. **Local variable doesn’t hold default value we have to initialize.**
   3. **Scope of variable within that block where it declare.**
3. **Static variable**

**Instance and local variable example**

class Car {

    int wheel;

    float price;

    String color;

    void start() {

        int temp=100;

        System.out.println("Car Start");

        System.out.println("Wheel "+wheel);

        System.out.println("Price "+price);

        System.out.println("Color "+color);

        System.out.println("Temp "+temp);

    }

    void stop() {

        String msg="Welcome";

        System.out.println("Car Stop");

        System.out.println("Wheel "+wheel);

        System.out.println("Price "+price);

        System.out.println("Color "+color);

        System.out.println("Msg "+msg);

    }

}

class CarTest {

    public static void main(String args[]){

        Car innova = new Car();     // heap memory

        innova.start();

        innova.stop();

    }

}