***Dt : 30/8/2022***

***2.Constructors with parameters:***

***=>The Constructors which are declared with parameters are***

***known as Parameterized constructors or Constructors with***

***parameters.***

***Ex-program : DemoCon2.java***

***class CTest2 //SubClass***

***{***

***CTest2(int x)//Constructor***

***{***

***System.out.println("====Constructor CTest(x)====");***

***System.out.println("The value x:"+x);***

***}***

***void dis(int y)//Instance method***

***{***

***System.out.println("====Instance dis(y)====");***

***System.out.println("The value y:"+y);***

***}***

***}***

***class DemoCon2 //MainClass***

***{***

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

***{***

***CTest2 ob = new CTest2(120);//Con\_call***

***ob.dis(123);//method\_call***

***}***

***}***

***o/p:***

***====Constructor CTest(x)====***

***The value x:120***

***====Instance dis(y)====***

***The value y:123***

***====================================================***

***Note:***

***=>we pass parameter to constructor while object creation***

***process because the Constructor\_Call is available in Object***

***creation syntax.***

***===================================================***

***\*imp***

***Advantage of Constructors:***

***=>Constructors are used to initialize instance variables***

***while object creation process and which saves the execution time***

***and generate HighPerformance of an application.***

***=====================================================***

***Ex-Program : DemoCon3.java***

***import java.util.Scanner;***

***class Employee //SubClass***

***{***

***//Instance Variable memory in Object***

***String id,name,desg;***

***int bSal;***

***float totSal;***

***//Constructor parameters are Local variables***

***//memory in Constructor***

***Employee(String id,String name,String desg,int bSal,***

***float totSal)***

***{***

***this.id=id;***

***this.name=name;***

***this.desg=desg;***

***this.bSal=bSal;***

***this.totSal=totSal;***

***}***

***//Getter method to display data from the object***

***void getEmployee()***

***{***

***System.out.println("====EmployeeDetails====");***

***System.out.println("EmpId:"+id);***

***System.out.println("EmpName:"+name);***

***System.out.println("EmpDesg:"+desg);***

***System.out.println("EmpBSal:"+bSal);***

***System.out.println("EmpTotSal:"+totSal);***

***}***

***}***

***class DemoCon3 //MainClass***

***{***

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

***{***

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

***//Local variables of main() method***

***System.out.println("Enter EmpID:");***

***String id = s.nextLine();***

***System.out.println("Enter EmpName:");***

***String name = s.nextLine();***

***System.out.println("Enter EmpDesg:");***

***String desg = s.nextLine();***

***System.out.println("Enter EmpbSal:");***

***int bSal = s.nextInt();***

***float totSal = bSal+(0.93F\*bSal)+(0.63F\*bSal);***

***Employee e = new Employee(id,name,desg,bSal,totSal);//Con\_Call***

***e.getEmployee();//method\_call***

***}***

***}***

***o/p:***

***Enter EmpID:***

***A111***

***Enter EmpName:***

***Raj***

***Enter EmpDesg:***

***SE***

***Enter EmpbSal:***

***15000***

***====EmployeeDetails====***

***EmpId:A111***

***EmpName:Raj***

***EmpDesg:SE***

***EmpBSal:15000***

***EmpTotSal:38400.0***

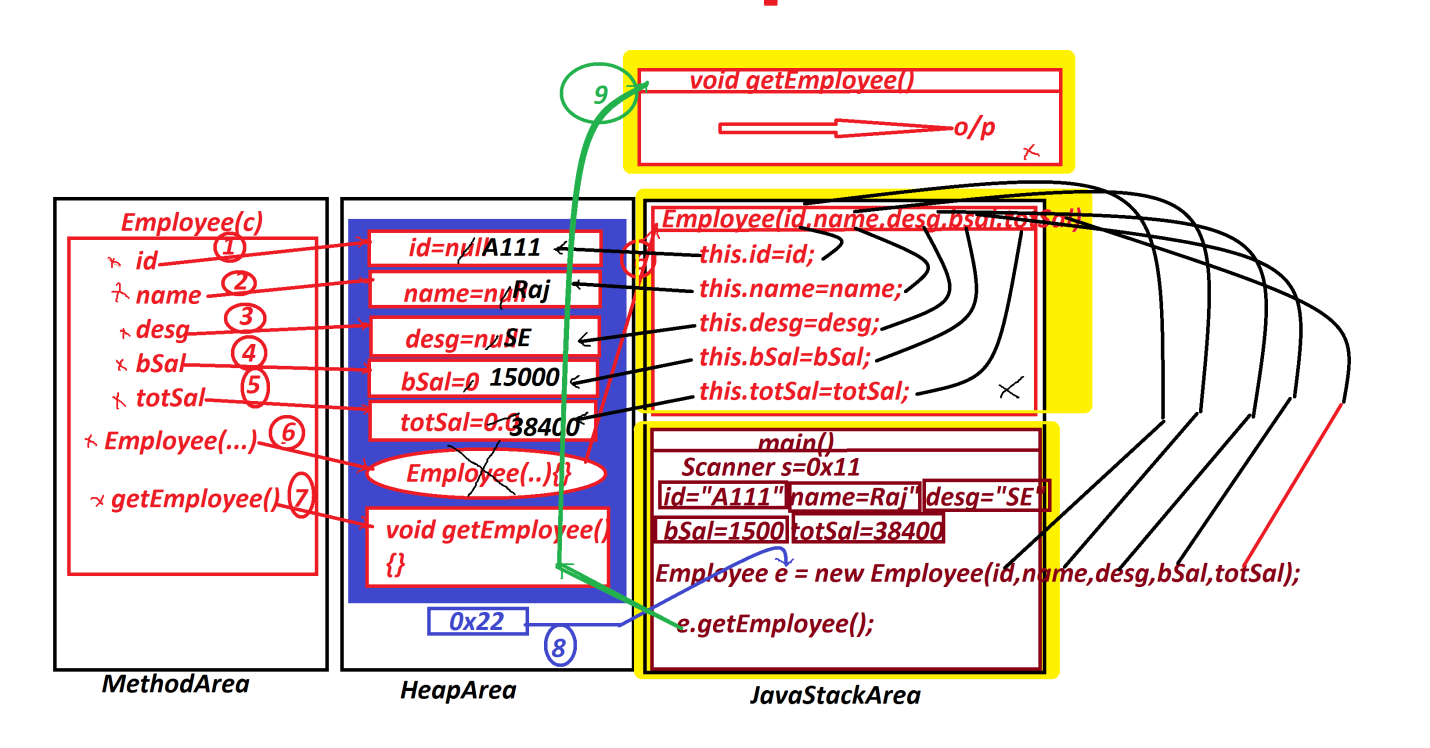
***=======================================================***

***Execution flow of above program:***

***ClassFiles:***

***Employee.class***

***DemoCon3.class(MainClass)***

******

***=======================================================***

***faq:***

***define "this" keyword?***

***=>"this" is a pre-defined Non-Primitive datatype variable***

***holding the reference of current running object.***

***=>using "this" variable we can access members from the object.***

***faq:***

***In wt situation we use "this" keyword?***

***=>when we load data from local variables to Instance variables***

***having same names,then we use "this" keyword.***

***======================================================***