13) Learn and Understand use of Constructors and its types and implemented the same for employee class with basic attribute as given below emp-name, emp-no, desig, basic-sal Instantiate employee object and calculate gross salary by providing appropriate earning and deductions.

```
-> package javaprogram;
   class Employee1 {
        String empName;
        int empNo;
        String empDesig;
        byte expYrs;
        double basicSalary;
        double netSalary;
        double grossSalary;
        double dearnessHallowance;
        double HRA;
        double PA;
        double IT;
        Employee1 ()
   {
      empName = null;
```

```
        empNo = 0;
        empDesign = null;
        expYrs = 0;
        basicSalary = 0.0d;
    }
Employee1 (String name, int no, String des,
    byte ex, double basicSalary) {
        empName = name;
        empNo = no;
        empDesig = des;
        expYrs = ex;
        basic Salary = basicSalary;
    }
Employee1 (Employee1 obj1  Employee1
        obj2) {
    empName = obj1.empName;
    empNo = obj2.empNo;
    emp Desig = obj1.empDesig;
    expYrs = obj2.expYrs;
    basicSalary = obj2.basic Salary;
}
void getEmployee() {
```

```
        deamessAllowance = basicSalary * 0.40;
        HRA = basicSalary * 0.25;
        PA = basicSalary * 1.0;
        grossSalary = basicSalary + deamessAllowance +
                      HRA + PA;
        IT = basicSalary * 1.0;
        netSalary = grossSalary - IT;
      System.out.println(empName + "\t" + empNo + "\t"+
      empDesig + "\t" + exprs + "\t" + basicSalary + "\t"+
       grossSalary + "\t\t" + netSalary);
      }
  }
public class grossSalary {
    public static void main (String[] args) {
  System.out.println("empName\t empNo\t empDesign\t
  exprs\t basicSalary\t grossSalary\t netSalary");
     Employee1 Rakesh = new Employee1();
     Rakesh.getEmployee();
     Employee1 Rahul = new Employee1("Rahul", 100
"PM", (byte) 10. 10000.0d);
     Rahul.getEmployee();
    Employee Ramya = new Employee (Rakesh, Rahul);
    Ramya.getEmployee();
    }
}
```

Output

Empname	Empno	EmpDesig	Expyrs	basicSalary	gross
Null	0	Null	0	0.0	
Rahul	100	PM	10	10000.0	

13.Learn and understand use of constructors and itstypes, and implement the same for employee class with basic attributes as given emp name,emp_no,desig,basic_sal. Instantiate employee objects and calculate gross salary by providing appropriate earnings and deductions.

```
class Employee1{
        String empName;
        int empNo;
        String empDesig;
        byte expYrs;
        double basicSalary;
        double netSalary;
        double grossSalary;
        double dearnessAllowance;
        double HRA;
        double PA;
        double IT;
        Employee1()
        {
                empName=null;
                empNo=0;
                empDesig=null;
                expYrs=0;
                basicSalary=0.0d;
        }
        Employee1(String name,int no,String des,byte ex,double basicsalary)
        {
                empName=name;
```

```
        empNo=no;
        empDesig=des;
        expYrs=ex;
    basicSalary=basicsalary;
}
Employee1(Employee1 obj1,Employee1 obj2)
{
        empName=obj1.empName;
        empNo=obj2.empNo;
        empDesig=obj1.empDesig;
        expYrs=obj2.expYrs;
        basicSalary=obj2.basicSalary;
}
void getEmployee1()
{
        dearnessAllowance=basicSalary*0.40;
        HRA=basicSalary*0.25;
        PA=basicSalary*10;
        grossSalary=basicSalary+dearnessAllowance+HRA+PA;

        IT=basicSalary*10;
        netSalary=grossSalary-IT;

System.out.println(empName+"\t"+empNo+"\t"+empDesig+"\t"+expYrs+"\t\t"+basicSalary+
"\t\t"+grossSalary+"\t\t"+netSalary);
    }
```

```
    public static void main(String[] args){

System.out.println("empName\tempNo\tempDesig\texpYrs\tbasicSalary\tgrossSalary\tnetS
alary");
            Employee1 rakesh=new Employee1();
            rakesh.getEmployee1();
            Employee1 rahul=new Employee1("Rahul",100,"PM",(byte)10,10000.0d);
            rahul.getEmployee1();
            Employee1 ramya=new Employee1(rakesh,rahul);
            ramya.getEmployee1();
    }

}
```

Output

empName	empNo	empDesig	expYrs	basicSalary	grossSalary	netSalary
null	0	null	0	0.0	0.0	0.0
Rahul	100	PM	10	10000.0	116500.0	16500.0
null	100	null	10	10000.0	116500.0	16500.0

14) Learn and Understand Autobox and unbox feauture in java and Design and implement a class for student with given attribute name, RegNo, Sem ia1, ia2, ia3 use constructor instantiate object implement method to calculate average marks of internal

```
Import Java.util.Scanner;
public class CalculateAverage {
    public static void main (String[] args){
        float average = 0.0f;
        Scanner sc = new Scanner (System.in);
        Integer Cie1, Cie2, Cie3;
        System.out.println ("Enter the Score of first
                Internal");
        Cie1 = Integer.parseInt (sc.next());
        System.out.println ("Enter the Score of Second
                Internal");
        Cie2 = Integer.parseInt (sc.next());
        System.out.println ("Enter the Score of Third
                Internal");
        Cie3 = Integer.parseInt (sc.next());
        average =(float) (Cie1 + Cie2 + Cie3)/3;
        System.out.println ("the average Internal
        Score is " + average);
        sc.close();
    }
}
```

Output

Enter the Score of Second Internal
19
Enter the Score of Third Internal
24
The average Internal Score is 23.33334

```
Import Java.util.Scanner;
Public class CalculateAverage{
 public static void main(String[] args)
 {
    float Sum=0.0f;
    Scanner Sc = new Scanner(System.in);
    Integer cie[] = new Integer[3];
  for (int i=0; i<3; i++){
  System.out.println("Enter the Score of
          Internal:" +(i+1));
    cie[i] = Integer.parseInt(Sc.next());
    Sum = Sum + cie[i];
  }
System.out.println("The average Inter
     -nal Score is" + (Sum/3));
  Sc.close();
} }
```

Output -

```
Enter the Score Internal : 1
16
Enter the Score of Internal : 2
161    38303    0
Enter the Score of Internal : 3
21
The average Internal Score is  22.33334
```

14.LEARN AND UNDERSTAND AUTOBOX AND UNBOX FEATURE IN JAVA AND DESIGN AND IMPLEMENT A CLASS FOR STUDENT WITH THE GIVEN ATTRIBUTES NAME,REGNO,SEM,IA1,IA2,IA3.USE CONSTRUCTORS TO INSTANTIATE OBJECTS.IMPLEMENT A METHOD TO CALCULATE AVERAGE MARKS OF INTERNALS.

```
import java.util.Scanner;
public class calculateAverage {
    public static void main(String[] args) {
        float average =0.0f;
        Scanner sc = new Scanner(System.in);

        /*This line declares array Integer Wrapper Object */
        Integer ciel, cie2, cie3;
        /* - Autobox
        * The parsed value of integer constant is converted wrapper
Object */
        System.out.println("Enter the score of First Internal:");
        ciel = Integer.parseInt(sc.next());
        System.out.println("Enter the score of Second Internal: ");
        cie2 = Integer.parseInt(sc.next());
        System.out.println("Enter the score of Third Internal: ");
        cie3= Integer.parseInt(sc.next());
        /* Unbox
        * The wrapper Object value is converted into Primitive Data
Type */
        average = (float) (ciel + cie2 + cie3)/3;
        System.out.println("The average Internal Score is "+average);
        sc.close();
    }
}
```

OUTPUT:

```
Enter the score of First Internal: 26
Enter the score of Second Internal: 15
Enter the score of Third Internal: 22
The average Internal Score is 21.
```

```
 /* This program illustrates Autobox and Unbox of Data Types */
import java.util.Scanner;
public class calculateAverage {
  public static void main(String[] args) {
    float sum =0.0f;
    Scanner sc = new Scanner (System.in);
    /*This line declares array Integer Wrapper Object */
    Integer cie[] = new Integer [3];

    for (int i=0; i< 3; i++) {
      System.out.println("Enter the score of Internal : "+(i+1));
      /* - Autobox
      * The parsed value of integer constant is converted wrapper
Object */
      cie[i] = Integer.parseInt(sc.next());
      /* Unbox
      * The wrapper Object value is converted into Primitive Data
Type */
      sum=sum + cie[i];
    }
      System.out.println("The average Internal Score is "+(sum/3));
      sc.close();

  }
}
```

OUTPUT:

```
Enter the score of Internal : 1
16
Enter the score of Internal : 2
161   38303    0
Enter the score of Internal : 3
21
The average Internal Score is 22.333334
```

15) Design and Implement a class for mobile store with given attribute make model category, price, discount, net_price use constructors to instantiate object and display details.

```
class MobileStore
{
    String make;
    String model;
    String category;
  double price;
  double discount;
  double netprice;
  MobileStore();
  {
     this.make = "Samsung";
     this.model = "Galaxy S21";
     this.category = "Smartphone";
     this.price = 80099.99;
     this.discount = 0.15;
  }
MobileStore(String make, String model, String cate
    -gory, double price, double discount) {
     This.make = make;
     This.model = model;
     This.category = category;
     This.price = price;
     This.discount = discount;
}
```

```
MobileStore (MobileStores obj. MobileStores obj1.
             MobileStore sobj2)
{
  make = sobj.make;
  model = sobj1. model;
  category = sobj2. category;
  price = sobj. price;
 discount = sobj2. discount;
}
boolean getDetails()
{
   System. out. println ("***");
   System. out. println (" company name: " + make);
   System. out. println ("model: " + model);
   System. out. println ("category: " + category);
   System. out. println (" price: " + price);
   System. out. println ("discount: " + discount);
   System. out. println (" net price: " + calculate
                netprice());
   System. out. println (" * * * ");
    return true;
}
double calculateNetPrice()
```

```
    {
        net price = price - (price * discount);
        return netprice;
    }
}

public class mobile
{
    public static void main (string[] args) {
        MobileStore mobile1 = new MobileStore ();
        mobile1.get Details ();
    MobileStore phone = new MobileStore ("Apple",
"iphone 12", "Smartphone", 99999.99 , 0.1);
    MobileStoreSamsung = new MobileStore ("Samsung",
"Galaxy S21", "Smartphone", 88899.99 , 0.15);
        iphone. get Details();
        Samsung. get Details ();
    MobileStore Goggle = new MobileStore ("Goggle",
"pixel 5", "Smartphone", 69999.99 , 02);
        Goggle. get Details ();
    MobileStore mobile 2 = new MobileStore (iphone,
        Samsung, Goggle);
        mobile 2. get Details ();

    }
}
```

Output

```
* * *
Company name - Samsung
 model - Galaxy S21
Category - Smartphone
 price - 80099.99
discount - 0.15
net price - 68084.9915
* * *
* * *
Company name - Apple
  model - iPhone 12
Category - Smartphone
 price - 99999.99
discount - 0.1
net price - 89999.991000000001
 * * *
 * * *

Company name - Samsung
  model - Galaxy S21
Category - Smartphone
 price - 88899.99
discount - 0.15
net price - 75564.9915
 * * *
 * * *
```

Company name - Goggle
model - pixel 5
category - Smartphone
price - 69999.99
discount - 0.2
net price - 55999.992000000000006
* * *
* * *
Company name - Tpple
model - Galaxy S21
category - Smartphone.
price - 99999.99
discount - 0.2
net price - 79999.992.
* * *

15.Design and implement a class for mobile store with given attributes make,model,category,price,discount,net_price.Use constructors to instantiate objects and display details.

```
class MobileStore
{
    String make;
    String model;
    String category;
    double price;
    double discount;
    double netPrice;

    MobileStore()
    {
        this.make="Samsung";
        this.model="Galaxy S21";
        this.category="Smartphone";
        this.price = 80099.99;
        this.discount = 0.15;
    }

    MobileStore(String make, String model, String category, double
price, double discount)
    {
        this.make = make;
        this.model = model;
        this.category = category;
        this.price = price;
        this.discount = discount;
    }

    MobileStore(MobileStore obj, MobileStore obj1, MobileStore obj2)
    {
        this.make = obj.make;
        this.model = obj1.model;
        this.category = obj2.category;
        this.price = obj.price;
        this.discount = obj2.discount;
    }

    boolean getDetails()
    {
        System.out.println("***");
        System.out.println("company name: "+make);
        System.out.println("model: "+model);
        System.out.println("category: "+category);
        System.out.println("price: "+price);
        System.out.println("discount: "+discount);
        System.out.println("net price: "+calculateNetPrice());
        System.out.println("***");
        return true;
    }
```

```
    double calculateNetPrice()
    {
        netPrice = price - (price* discount);
        return netPrice;
    }
}

public class mobile
{
    public static void main(String args[])
    {
        MobileStore mobile1 = new MobileStore();
        mobile1.getDetails();

        MobileStore iphone = new MobileStore("Apple", "iPhone 12",
"Smartphone", 99999.99, 0.1);
        iphone.getDetails();

        MobileStore Samsung = new MobileStore("Samsung", "Galaxy
S21", "Smartphone", 88899.99, 0.15);
        Samsung.getDetails();

        MobileStore Google = new MobileStore("Google", "Pixel 5",
"Smartphone", 69999.99, 0.2);
        Google.getDetails();

        MobileStore mobile2 = new MobileStore(iphone,Samsung,Google);
        mobile2.getDetails();
    }


}
```

Ouput:

```
***

company name: Samsung

model: Galaxy S21

category: Smartphone

price: 80099.99

discount: 0.15

net price: 68084.9915

***

***

company name: Apple

model: iPhone 12
```

```
company name: Samsung
model: Galaxy S21
category: Smartphone
price: 80099.99
discount: 0.15
net price: 68084.9915
***
***
company name: Apple
model: iPhone 12
```

```
category: Smartphone
price: 99999.99
discount: 0.1
net price: 89999.99100000001
***
***
company name: Samsung
model: Galaxy S21
category: Smartphone
price: 88899.99
discount: 0.15
net price: 75564.9915
***
***
company name: Google
model: Pixel 5
category: Smartphone
price: 69999.99
discount: 0.2
net price: 55999.992000000006
***
***
company name: Apple
model: Galaxy S21
category: Smartphone
price: 99999.99
discount: 0.2
net price: 79999.992
***
```