**1st OBJECTIVE:**

Create a class Point, with two properties x and y. Write all the methods to manipulate the values of Point. Write a method that can check two Objects of Point class for Equality.

**PROGRAM # 1:**

SOURCE CODE:

class point

{

int x, y;

point(int i, int j)

{

x = i;

y = j;

}

boolean equals(point o)

{

if(o.x == x && o.y == y)

return true;

else

return false;

} }

class pass

{

public static void main(String args[])

{

point ob1 = new point(100, 22);

point ob2 = new point(100, 22);

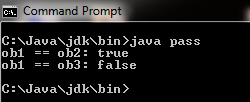
point ob3 = new point(2,1);

System.out.println("ob1 == ob2: " + ob1.equals(ob2));

System.out.println("ob1 == ob3: " + ob1.equals(ob3));

}}

OUTPUT:



CONCLUSION:

In this program we are learning how to use method.

**2nd OBJECTIVE:**

Designed a class called circle and it is contains:

Two private instance variables: radius (of the type double) and color (of the type String), with default value of 1.0 and "red", respectively.

Two *overloaded* constructors - a *default* constructor with no argument, and a constructor which takes a double argument for radius.

Two public methods: getRadius() and getArea(), which return the radius and area of this instance, respectively.

**PROGRAM # 2:**

SOURCE CODE:

class circle

{

private double radius;

private String colour;

public circle()

{

radius=1.0;

colour="red";

}

public circle(double r)

{

radius=r;

colour="red";

}

public double getRadius()

{

return radius;

}

public double getArea()

{

return radius\*radius\*Math.PI;

}

}

public class testcir

{

public static void main(String[] args)

{

circle obj=new circle();

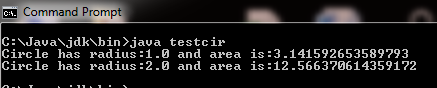
System.out.println("Circle has radius:"+obj.getRadius()+" and area is:"+obj.getArea());

circle obj2=new circle(2.0);

System.out.println("Circle has radius:"+obj2.getRadius()+" and area is:"+obj2.getArea());

}}

OUTPUT:



CONCLUSION:

In this program we are learning how to over load constructer.

**3rd OBJECTIVE:**

Create a class called Date that includes three pieces of information as instance variables—a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date’s capabilities.

**PROGRAM # 3:**

SOURCE CODE:

if(m>0 && m<=12)

month = m;

else month = 1;

if(d>0 && d<=30)

day = d;

else day = 1;

}

public int getYear()

{

return year;

}

public void setYear(int y)

{

if(y>=0)

year = y;

else year = 2000;

}

public int getMonth()

{

return month;

}

public void setMonth(int m)

{

if(m>0 && m<=12)

month = m;

else month = 1;

}

public int getDay()

{

return day;

}

public void setDay(int d)

{

if(d>0 && d<=30)

day = d;

else day = 1;

}

public void display()

{

System.out.println(day+"/"+month+"/"+year);

}}

public class tstdate

{

public static void main(String [] args)

{

Scanner S = new Scanner(System.in);

System.out.print("Enter year: ");

int year = S.nextInt();

System.out.print("Enter month: ");

int month = S.nextInt();

System.out.print("Enter day: ");

int day = S.nextInt();

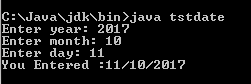
Date date = new Date(year, month, day);

System.out.print("You Entered :");

date.display();

}}

OUTPUT:



CONCLUSION:

In this program we are learning how to use get and set methods.

**4th OBJECTIVE:**

create a class called Employee that includes three pieces of information as instance variables—a first name (type String), a last name (type String) and a monthly salary (type double). Your class should have a constructor that initializes the three instance variables. Provide a set and a get method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates class Employee’s capabilities. Create two Employee objects and display the yearly salary for each Employee. Then give each Employee a 10% raise and display each Employee’s yearly salary again.

**PROGRAM # 4:**

SOURCE CODE:

import java.util.Scanner;

class Employee

{

private String firstName;

private String lastName;

private double monthlySalary;

public Employee(String f, String l, double m)

{

firstName = f;

lastName = l;

if(m < 0)

{

monthlySalary =0;

}

else monthlySalary = m;

}

public String getFirstName()

{

return firstName;

}

public void setFirstName(String fname)

{

firstName = fname;

}

public String getLastName()

{

return lastName;

}

public void setLastName(String lname)

{

lastName = lname;

}

public double getMonthlySalary()

{

return monthlySalary;

}

public void setMonthlySalary(double m)

{

if(m < 0)

{

monthlySalary =0;

}

else monthlySalary = m;

}

}

public class emptest

{

public static void main(String[] args)

{

Scanner S = new Scanner(System.in);

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

String fname = S.next();

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

String lname = S.next();

System.out.println("Enter the Salary: ");

double sal = S.nextDouble();

Employee e =new Employee(fname,lname ,sal );

System.out.println("the yearly salary of "+e.getFirstName()+" "+e.getLastName()+" :");

System.out.println(e.getMonthlySalary()\*12);

double newsalary= e.getMonthlySalary()\*0.1+e.getMonthlySalary();

e.setMonthlySalary(newsalary);

System.out.println("the new yearly salary of "+

e.getFirstName()+" "+e.getLastName()+" :");

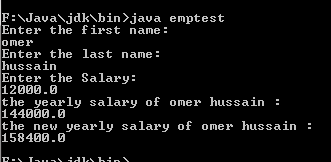
System.out.println(e.getMonthlySalary()\*12);

e.getMonthlySalary();

}

}

OUTPUT:



CONCLUSION:

In this program we are learning how to use get and set method and constructor.

**5th OBJECTIVE:**

Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information as instance variables—a part number (type String), a part description (type String), a quantity of the item being purchased (type int) and a price per item (double). Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a method named getInvoiceAmount that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as a double value. If the quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to 0.0. Write a test application named InvoiceTest that demonstrates class Invoice’s capabilities.

**PROGRAM # 5:**

SOURCE CODE:

class Invoice

{

private String partNumber;

private String partDescription;

private int quantity;

private double pricePerItem;

public Invoice( String part, String description, int count,double price )

{

partNumber = part;

partDescription = description;

if ( count > 0 ) quantity = count;

if ( price > 0.0 )

pricePerItem = price;

}

public void setPartNumber( String part )

{

partNumber = part;

}

public String getPartNumber()

{

return partNumber;

}

public void setPartDescription( String description )

{

partDescription = description;

}

public String getPartDescription()

{

return partDescription;

}

public void setQuantity( int count )

{

if ( count > 0 )

quantity = count;

if ( count <= 0 )

quantity = 0;

}

public int getQuantity()

{

return quantity;

}

public void setPricePerItem( double price )

{

if ( price > 0.0 ) pricePerItem = price;

if ( price <= 0.0 )

pricePerItem = 0.0; }

public double getPricePerItem()

{

return pricePerItem;

}

public double getInvoiceAmount()

{

return getQuantity() \* getPricePerItem();

}

}

public class InvoiceTest

{

public static void main( String args[] )

{

Invoice invoice1 = new Invoice( "1234", "Hammer", 2, 14.95 );

System.out.println( "Original invoice information" );

System.out.printf( "Part number: %s\n", invoice1.getPartNumber() );

System.out.printf( "Description: %s\n",

invoice1.getPartDescription() );

System.out.printf( "Quantity: %d\n", invoice1.getQuantity() );

System.out.printf( "Price: %.2f\n", invoice1.getPricePerItem() );

System.out.printf( "Invoice amount: %.2f\n",

invoice1.getInvoiceAmount() );

invoice1.setPartNumber( "001234" );

invoice1.setPartDescription( "Yellow Hammer" );

invoice1.setQuantity( 3 );

invoice1.setPricePerItem( 19.49 );

System.out.println( "\nUpdated invoice information" );

System.out.printf( "Part number: %s\n", invoice1.getPartNumber() );

System.out.printf( "Description: %s\n",

invoice1.getPartDescription() );

System.out.printf( "Quantity: %d\n", invoice1.getQuantity() );

System.out.printf( "Price: %.2f\n", invoice1.getPricePerItem() );

System.out.printf( "Invoice amount: %.2f\n",

invoice1.getInvoiceAmount() );

Invoice invoice2 = new Invoice( "5678", "Paint Brush", -5, -9.99 );

System.out.println( "\nOriginal invoice information" );

System.out.printf( "Part number: %s\n", invoice2.getPartNumber() );

System.out.printf( "Description: %s\n",

invoice2.getPartDescription() );

System.out.printf( "Quantity: %d\n", invoice2.getQuantity() );

System.out.printf( "Price: %.2f\n", invoice2.getPricePerItem() );

System.out.printf( "Invoice amount: %.2f\n",

invoice2.getInvoiceAmount() );

invoice2.setQuantity( 3 );

invoice2.setPricePerItem( 9.49 );

System.out.println( "\nUpdated invoice information" );

System.out.printf( "Part number: %s\n", invoice2.getPartNumber() );

System.out.printf( "Description: %s\n",

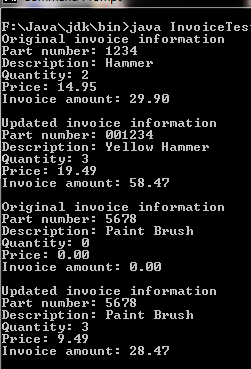
invoice2.getPartDescription() );

System.out.printf( "Quantity: %d\n", invoice2.getQuantity() );

System.out.printf( "Price: %.2f\n", invoice2.getPricePerItem() );

System.out.printf( "Invoice amount: %.2f\n",invoice2.getInvoiceAmount() ); } }

OUTPUT:



CONCLUSION:

In this program we are learning how to use get and set method and constructor.

**6th OBJECTIVE:**

Create a class with a name Calculate. Add two basic arithmetic functions to it, such as add() and subtract() to perform mathematical calculations. Now overload these methods so that they can take three types of values, an int, a double or a char. Note that if characters are passed to a method, it should return char, if double is sent to a method it should give its answer in double, and so on.

**PROGRAM # 6:**

SOURCE CODE:

class cal{

int add(int a,int b)

{

int c;

c=a+b;

return c;

}

double add(double a,double b)

{

double c;

c=a+b;

return c;

}

int add(char a,char b)

{

int c;

c=(int)a+(int)b;

return c;

}

int sub(int a,int b)

{

int c;

c=a-b;

return c;

}

double sub(double a,double b)

{

double c;

c=a-b;

return c;

}

int sub(char a,char b)

{

int c;

c=(int)a-(int)b;

return c;

}

}

public class load

{

public static void main(String[] args)

{

cal obj=new cal();

System.out.println("add int:"+obj.add(10,10));

System.out.println("add double:"+obj.add(12.3,12.7));

System.out.println("add char:"+(char)obj.add('A','a'));

System.out.println("Subtract int:"+obj.sub(15,10));

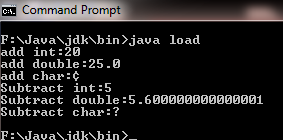
System.out.println("Subtract double:"+obj.sub(18.3,12.7));

System.out.println("Subtract char:"+(char)obj.sub('A','g'));

}

}

OUTPUT:



CONCLUSION:

In this program we are learning how to overload constructor.

**7th OBJECTIVE:**

Create a Class Line. Line Objects will be created by 2 Point Objects. Create 2 Line objects and check their equality by calling Point class equals function from Line class equals function.

**PROGRAM # 7:**

SOURCE CODE:

class line

{

int x, y;

line(int i,int j)

{

x=i;

y=j;

}

boolean equals(line a)

{

if(a.x==x && a.y==y)

return true;

else

return false;

}

}

public class exline

{

public static void main(String[] args)

{

line ob1=new line(11,29);

line ob2=new line(11,29);

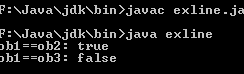
line ob3=new line(1,9);

System.out.println("ob1==ob2: " + ob1.equals(ob2));

System.out.println("ob1==ob3: " + ob1.equals(ob3));

}}

OUTPUT:



CONCLUSION:

In this program we are learning how check values are equal or not.