**Assignment 1:**

**1:write program to test Hello World.**

**public** **class** Hello {

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

// **TODO** Auto-generated method stub

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

}

}

**Output:**

Hello World

**2:Write a program to adddition of two numbers .**

**import** java.util.Scanner;

**public** **class** Addition {

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

// **TODO** Auto-generated method stub

**int** num1,num2,c;

System.***out***.println("Enter Two numbers");

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

num1=sc.nextInt();

num2=sc.nextInt();

c=num1+num2;

System.***out***.println("Addition="+c);

}

}

**Output:**

Enter Two numbers

45

66

Addition=111

**3:Write a program to swap two numbers**.

**import** java.util.Scanner;

**public** **class** Swap {

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

// **TODO** Auto-generated method stub

**int** a,b;

System.***out***.println("Enter Two numbers");

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

a=sc.nextInt();

b=sc.nextInt();

System.***out***.println("Before Swap "+a+" "+b);

a=a+b;

b=a-b;

a=a-b;

System.***out***.println("After Swap "+a+" "+b);

}

}

**Output:**

Enter Two numbers

23

12

Before Swap 23 12

After Swap 12 23

**4:Write a program to find factorial of a given number**.

**import** java.util.Scanner;

**public** **class** Factorial {

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

// **TODO** Auto-generated method stub

**int** n,i,res=1;

System.***out***.println("Enter number");

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

n=sc.nextInt();

**for**(i=1;i<=n;i++)

{

res=res\*i;

}

System.***out***.println("Factorial="+res);

}

}

**Output:**

Enter number

5

Factorial=120

**5:Write a program to find m to the power n.**

**import** java.util.Scanner;

**public** **class** Power {

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

// **TODO** Auto-generated method stub

**int** m,n,i,res=1;

System.***out***.println("Enter number");

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

m=sc.nextInt();

System.***out***.println("Enter power");

n=sc.nextInt();

**for**(i=1;i<=n;i++)

{

res=res\*m;

}

System.***out***.println("Result"+res);

}

}

**Output:**

Enter number

4

Enter power

3

Result=64

**6:Check if number is a prime number or not.**

**import** java.util.Scanner;

**public** **class** Test1 {

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

// **TODO** Auto-generated method stub

**int** n,i,count=0;

System.***out***.println("Enter number");

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

n=sc.nextInt();

**for**(i=1;i<=n;i++)

{

**if**(n%i==0)

{

count++;

}

}

**if**(count==2)

{

System.***out***.println("It is Prime number");

}

**else**

{

System.***out***.println("It is not Prime number");

}

}

}

**Output:**

Enter number

45

It is not Prime number

Enter number

23

It is Prime number

**7:Sum of series :**

**1+2+3+….+n**

**import** java.util.Scanner;

**public** **class** Sum {

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

// **TODO** Auto-generated method stub

**int** n,i,sum=0;

System.***out***.println("Enter number");

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

n=sc.nextInt();

**for**(i=1;i<=n;i++)

{

sum=sum+i;

}

System.***out***.println("Sum="+sum);

}

}

**Output:**

Enter number

10

Sum=55

**8:Write a program to find sum of all even and odd numbers between 1 to n.**

**import** java.util.Scanner;

**public** **class** EvenOddSum {

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

// **TODO** Auto-generated method stub

**int** n,i,sum=0,a=0;

System.***out***.println("Enter number");

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

n=sc.nextInt();

**for**(i=1;i<=n;i++)

{

**if**(i%2==0)

{

sum=sum+i;

}

**else**

{

a=a+i;

}

}

System.***out***.println("even sum"+sum);

System.***out***.println("odd sum"+a);

}

}

**Output:**

Enter number

10

even sum=30

odd sum=25

**10: Write a program to enter a number and print its reverse.**

**import** java.util.Scanner;

**public** **class** Reverse {

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

// **TODO** Auto-generated method stub

**int** n,i,rev=0,temp,r;

System.***out***.println("Enter number");

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

n=sc.nextInt();

temp=n;

**while**(temp!=0)

{

r=temp%10;

rev=rev\*10+r;

temp=temp/10;

}

System.***out***.println("Reverse="+rev);

}

}

**Output:**

Enter number

123

Reverse=321

**11:Write a program to print all Prime numbers between 1 to n.**

**import** java.util.Scanner;

**public** **class** Prime {

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

// **TODO** Auto-generated method stub

**int** n,i,j,count;

System.***out***.println("Enter number");

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

n=sc.nextInt();

System.***out***.println("Prime numbers are");

**for**(i=1;i<=n;i++)

{

count=0;

**for**(j=1;j<=i;j++)

{

**if**(i%j==0)

{

count++;

}

}

**if**(count==2)

{

System.***out***.println(i);

}

}

}

}

**Output:**

Enter number

10

Prime numbers are

2

3

5

7

**12:Write a program to check entered number is Armstrong number or not.**

**import** java.util.Scanner;

**public** **class** Armstrong {

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

// **TODO** Auto-generated method stub

**int** n,res=0,r,temp;

System.***out***.println("Enter number");

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

n=sc.nextInt();

temp=n;

**while**(temp!=0)

{

r=temp%10;

res=res+(r\*r\*r);

temp=temp/10;

}

**if**(res==n)

{

System.***out***.println("It is Armstrong number");

}

**else**

{

System.***out***.println("It is not Armstrong number");

}

}

}

**Output:**

Enter number

153

It is Armstrong number

Enter number

121

It is not Armstrong number

**13:Write a program to find greatest of three numbers using nested if-else.**

**import** java.util.Scanner;

**public** **class** MaximumNumber {

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

// **TODO** Auto-generated method stub

**int** num1,num2,num3;

System.***out***.println("Enter three numbers");

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

num1=sc.nextInt();

num2=sc.nextInt();

num3=sc.nextInt();

**if**(num1<num2)

{

**if**(num2<num3)

{

System.***out***.println(num3+" is maximum");

}

**else**

{

System.***out***.println(num2+" is maximum");

}

}

**else**

{

**if**(num1<num3)

{

System.***out***.println(num3+" is maximum");

}

**else**

{

System.***out***.println(num1+" is maximum");

}

}

}

}

**Output:**

Enter three numbers

12

45

34

45 is maximum

**14:Create menu driven program for Pizza Shop.And display total amount,**

**import** java.util.Scanner;

**public** **class** PizzaShop {

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

// **TODO** Auto-generated method stub

**int** choice;

**double** amount=0;

System.***out***.println("1.Cheese Pizza 2.Pepperoni Pizza 3.Meat Pizza 4.Margherita Pizza 5.Exit");

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

**do** {

System.***out***.println("Enter choice");

choice=sc.nextInt();

**switch**(choice)

{

**case** 1:

System.***out***.println("Enter Qty for Cheese Pizza");

**int** qty=sc.nextInt();

amount=amount+(qty\*100);

System.***out***.println("Total Amount="+amount);

**break**;

**case** 2:

System.***out***.println("Enter Qty for Pepperoni Pizza");

qty=sc.nextInt();

amount=amount+(qty\*120);

System.***out***.println("Total Amount="+amount);

**break**;

**case** 3:

System.***out***.println("Enter Qty for Meat Pizza");

qty=sc.nextInt();

amount=amount+(qty\*140);

System.***out***.println("Total Amount="+amount);

**break**;

**case** 4:

System.***out***.println("Enter Qty for Margherita Pizza");

qty=sc.nextInt();

amount=amount+(qty\*160);

System.***out***.println("Total Amount="+amount);

**break**;

**case** 5:

System.***out***.println("Total Amount="+amount);

**break**;

**default**:

System.***out***.println("Invalid choice");

**break**;

}

}**while**(choice>0 && choice<6);

System.***out***.println("Exit");

sc.close();

}

}

**Output:**

1.Cheese Pizza 2.Pepperoni Pizza 3.Meat Pizza 4.Margherita Pizza 5.Exit

Enter choice

1

Enter Qty for Cheese Pizza

2

Total Amount=200.0

Enter choice

2

Enter Qty for Pepperoni Pizza

3

Total Amount=560.0

Enter choice

3

Enter Qty for Meat Pizza

4

Total Amount=1120.0

Enter choice

4

Enter Qty for Margherita Pizza

6

Total Amount=2080.0

Enter choice

5

Total Amount=2080.0

Enter choice

7

Invalid choice

Exit

**15:Create Menu driven program for array operations**

**import** java.util.Scanner;

**public** **class** MenuDrivenArray {

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

**int** ch, size;

System.***out***.println(

" 1:Read Array\n 2:Print Array \n 3:Search element in array\n 4:Reverse Array\n 5:Even number from array\n 6:sum of array element");

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

System.***out***.println("Enter size of Array ");

size = sc.nextInt();

**int** arr[] = **new** **int**[size];

System.***out***.println("Array created for length " + arr.length);

**do** {

System.***out***.println("Enter Choice ");

ch = sc.nextInt();

**switch** (ch) {

**case** 1:

System.***out***.println("Enter Elements of array ");

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

arr[i] = sc.nextInt();

}

**break**;

**case** 2:

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

System.***out***.println(" " + arr[i]);

}

**break**;

**case** 3:

System.***out***.println("Enter element to search");

**int** no = sc.nextInt();

**boolean** flag = **false**;

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

**if** (arr[i] == no) {

flag = **true**;

**break**;

} **else**

flag = **false**;

}

**if** (flag == **true**)

System.***out***.println("Entered element is found in array");

**else**

System.***out***.println("Not found in array");

**break**;

**case** 4:

System.***out***.println("Reverse Array is ");

**for** (**int** i = size - 1; i >= 0; i--) {

System.***out***.println(" " + arr[i]);

}

Break;

**case** 5:

System.***out***.println("Even no. in Array are ");

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

**if** (arr[i] % 2 == 0)

System.***out***.println(arr[i]);

}

**break**;

**case** 6:

**int** total = 0;

System.***out***.println("Sum of Element in Array are ");

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

total = total + arr[i];

}

System.***out***.println(total);

**break**;

**default**:

System.***out***.println("invalid choice");

**break**;

}

} **while** (ch <= 6);

sc.close();

}

}

Output:

1:Read Array

2:Print Array

3:Search element in array

4:Reverse Array

5:Even number from array

6:sum of array element

Enter size of Array

5

Array created for length 5

Enter Choice

1

Enter Elements of array

12

23

34

45

67

Enter Choice

2

12

23

34

45

67

Enter Choice

3

Enter element to search

67

Entered element is found in array

Enter Choice

4

Reverse Array is

67

45

34

23

12

Enter Choice

5

Even no. in Array are

12

34

Enter Choice

6

Sum of Element in Array are

181

Enter Choice

7

invalid choice

**16:read two int array...and store both in third array and display third array**

**1 2 3**

**5 6 7 8 9**

**1 2 3 5 6 7 8 9**

**import** java.util.Scanner;

**public** **class** Array {

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

// **TODO** Auto-generated method stub

**int** i, size1, size2;

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

System.***out***.println("Enter First Array size");

size1 = sc.nextInt();

System.***out***.println("Enter Second Array size");

size2 = sc.nextInt();

**int** arr1[] = **new** **int**[size1];

**int** arr2[] = **new** **int**[size2];

System.***out***.println("Enter First Array elements");

**for** (i = 0; i < arr1.length; i++) {

arr1[i] = sc.nextInt();

}

**int** a1 = arr1.length;

System.***out***.println("Enter Second Array elements");

**for** (i = 0; i < arr2.length; i++) {

arr2[i] = sc.nextInt();

}

**int** a2 = arr2.length;

**int** arr3[] = **new** **int**[a1 + a2];

**for** (i = 0; i < a1; i++) {

arr3[i] = arr1[i];

}

**for** (**int** j = 0; j < a2; j++) {

arr3[a1 + j] = arr2[j];

}

sc.close();

System.***out***.print("Final Array");

**for** (i = 0; i < arr3.length; i++) {

System.***out***.print(" "+arr3[i]);

}

}

}

**Output:**

Enter First Array size

3

Enter Second Array size

6

Enter First Array elements

1

2

3

Enter Second Array elements

4

5

6

7

8

9

Final Array 1 2 3 4 5 6 7 8 9

**Assignment 2:**

**1:Create Date class with members day,month ,year.**

**Write no argument and parameterised constructor .Create two object s and initialize them using no argument and parameterised constructor**

**respectively.Print date using display function.**

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** Date {

**int** day,month,year;

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

**public** Date()

{

System.***out***.println("----Inside Parameterless Constructor-----");

**this**.day=sc.nextInt();

**this**.month=sc.nextInt();

**this**.year=sc.nextInt();

}

**public** Date(**int** day,**int** month,**int** year)

{

**this**.day=day;

**this**.month=month;

**this**.year=year;

}

**public** **void** acceptinfo()

{

day=sc.nextInt();

month=sc.nextInt();

year=sc.nextInt();

}

**public** **void** display()

{

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

}

}

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** DateTester {

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

// **TODO** Auto-generated method stub

Date d1,d2;

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

System.***out***.println("D1 Object");

d1=**new** Date();

d1.display();

System.***out***.println("----Inside Parameterised Constructor-----");

d1=**new** Date(sc.nextInt(),sc.nextInt(),sc.nextInt());

// d1=new Date(11,2,2000);

d1.display();

System.***out***.println("D2 Object");

d2=**new** Date();

d2.display();

System.***out***.println("----Inside Parameterised Constructor-----");

d2=**new** Date(sc.nextInt(),sc.nextInt(),sc.nextInt());

d2.display();

sc.close();

}

}

**Output:**

D1 Object

----Inside Parameterless Constructor-----

12

2

2000

12/2/2000

----Inside Parameterised Constructor-----

23

12

2000

23/12/2000

D2 Object

----Inside Parameterless Constructor-----

2

2

2000

2/2/2000

----Inside Parameterised Constructor-----

1

4

10

1/4/10

**2:Create Employee class with members id(int),name(string),dob(Date).Use above created Date class.**

**Write default and parameterised constructor in Employee Class.Write accept() function to accept information and display() to display emp information.**

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** Employee {

**int** empid;

String name;

Date date;

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

**public** Employee()

{

System.***out***.println("----Inside Default Constructor-----");

empid=sc.nextInt();

name=sc.next();

date= **new** Date();

}

**public** Employee(**int** empid,String name,Date date)

{

**this**.empid=empid;

**this**.name=name;

**this**.date=date;

}

**public** **void** acceptinfo()

{

System.***out***.println("Accept info");

empid=sc.nextInt();

name=sc.next();

date.acceptinfo();

}

**public** **void** displayinfo(){

{

System.***out***.println("Employee ID="+empid);

System.***out***.println("Employee name="+name);

System.***out***.print("Employee Joining Date");

date.display();

}

}

}

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** EmployeeTester {

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

// **TODO** Auto-generated method stub

Employee e1;

//d1=new Date();

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

e1=**new** Employee();

e1.displayinfo();

System.***out***.println("----Inside Parameterised Constructor-----");

e1=**new** Employee(sc.nextInt(),sc.next(),**new** Date());

e1.displayinfo();

e1.acceptinfo();

e1.displayinfo();

sc.close();

}

}

**Output:**

----Inside Default Constructor-----

01

Abhijeet

03

02

2022

Employee ID=1

Employee name=Abhijeet

Employee Joining Date3/2/2022

----Inside Parameterised Constructor-----

2

Pranav

04

02

2022

Employee ID=2

Employee name=Pranav

Employee Joining Date4/2/2022

Accept info

03

Aniket

04

02

2022

Employee ID=3

Employee name=Aniket

Employee Joining Date4/2/2022

**3. Create a class Person with data members as name, age, city. Write getters and setters for all the data**

**members. Also add the display function. Create Default and Parameterized constructors. Create the**

**object of this class in main method and invoke all the methods in that class.**

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** Person {

String name,city;

**int** age;

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

**public** Person()

{

System.***out***.println("----Inside Parameterless Constructor----");

name=sc.next();

age=sc.nextInt();

city=sc.next();

}

**public** Person(String name,**int** age,String city)

{

**this**.name=name;

**this**.age=age;

**this**.city=city;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name =name;

}

**public** String getCity() {

**return** city;

}

**public** **void** setCity(String city) {

**this**.city = city;

}

**public** **int** getAge() {

**return** age;

}

**public** **void** setAge(**int** age) {

**this**.age = age;

}

**public** **void** display()

{

System.***out***.println("Name="+name+" Age="+age+" City="+city);

}

}

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** PersonTester {

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

// **TODO** Auto-generated method stub

Person p1;

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

p1=**new** Person();

p1.display();

System.***out***.println("----Inside Parameterised Constructor----");

p1=**new** Person(sc.next(),sc.nextInt(),sc.next());

p1.display();

p1.setName(sc.next());

p1.getName();

p1.display();

p1.setAge(sc.nextInt());

p1.getAge();

p1.display();

p1.setCity(sc.next());

p1.getCity();

p1.display();

sc.close();

}

}

**Output:**

----Inside Parameterless Constructor----

Ketan

23

Pune

Name=Ketan Age=23 City=Pune

----Inside Parameterised Constructor----

Suyash

23

Sangli

Name=Suyash Age=23 City=Sangli

Anil

Name=Anil Age=23 City=Sangli

25

Name=Anil Age=25 City=Sangli

Satara

Name=Anil Age=25 City=Satara

**4: Create a class Book with data members as bname,id,author,price. Write getters and setters for all the data members. Also add the display function.**

**Create the object of this class in main method and invoke all the methods in that class.**

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** Book {

String bname,author;

**int** id;

**double** price;

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

**public** String getBname() {

**return** bname;

}

**public** **void** setBname(String bname) {

**this**.bname = bname;

}

**public** String getAuthor() {

**return** author;

}

**public** **void** setAuthor(String author) {

**this**.author = author;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** **double** getPrice() {

**return** price;

}

**public** **void** setPrice(**double** price) {

**this**.price = price;

}

**public** **void** display()

{

System.***out***.println("Book Name="+bname+",Book ID="+id+",Author Name="+author+",Book Price="+price);

}

}

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** BookTester {

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

// **TODO** Auto-generated method stub

Book b1;

b1=**new** Book();

b1.display();

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

System.***out***.println("Set Book Name");

b1.setBname(sc.next());

b1.getBname();

System.***out***.println("Set Book ID");

b1.setId(sc.nextInt());

b1.getId();

System.***out***.println("Set Book Author");

b1.setAuthor(sc.next());

b1.getAuthor();

System.***out***.println("Set Book Price");

b1.setPrice(sc.nextDouble());

b1.getPrice();

b1.display();

sc.close();

}

}

**Output:**

Book Name=null,Book ID=0,Author Name=null,Book Price=0.0

Set Book Name

LetUsJava

Set Book ID

01

Set Book Author

YashwantKanetkar

Set Book Price

400

Book Name=LetUsJava,Book ID=1,Author Name=YashwantKanetkar,Book Price=400.0

**5. Create a class Point with data members as x,y. Write**

**getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class**.

**package** Day2Assignment;

**public** **class** Point {

**int** x,y;

**public** **int** getX() {

**return** x;

}

**public** **void** setX(**int** x) {

**this**.x = x;

}

**public** **int** getY() {

**return** y;

}

**public** **void** setY(**int** y) {

**this**.y = y;

}

**public** **void** display()

{

System.***out***.println("First Point="+x+",Second Point="+y);

System.***out***.println("Point("+x+","+y+")");

}

}

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** PointTester {

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

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

Point p1;

p1=**new** Point();

p1.display();

System.***out***.println("Set X");

p1.setX(sc.nextInt());

p1.getX();

System.***out***.println("Set Y");

p1.setY(sc.nextInt());

p1.getY();

p1.display();

sc.close();

}

}

**Output:**

First Point=0,Second Point=0

Point(0,0)

Set X

2

Set Y

3

First Point=2,Second Point=3

Point(2,3)

**6. Create a class ComplexNumber with data members real, imaginary. Write getters and setters for all the data members. Also add the display function. Create the object of this class in main method and invoke all the methods in that class.**

**package** Day2Assignment;

**public** **class** Complex {

**int** real,imag;

**public** **int** getReal() {

**return** real;

}

**public** **void** setReal(**int** real) {

**this**.real = real;

}

**public** **int** getImag() {

**return** imag;

}

**public** **void** setImag(**int** imag) {

**this**.imag = imag;

}

**public** **void** display()

{

System.***out***.println("Real="+real+",Imag="+imag);

System.***out***.println("C1("+real+","+imag+")");

}

}

**package** Day2Assignment;

**import** java.util.Scanner;

**public** **class** ComplexTester {

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

// **TODO** Auto-generated method stub

Complex c1;

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

c1=**new** Complex();

c1.display();

System.***out***.println("Set Real");

c1.setReal(sc.nextInt());

c1.getReal();

System.***out***.println("Set Imag");

c1.setImag(sc.nextInt());

c1.getImag();

c1.display();

sc.close();

}

}

**Output:**

Real=0,Imag=0

C1(0,0)

Set Real

3

Set Imag

5

Real=3,Imag=5

C1(3,5)

**7:create BankAccount aaplication for operations like withdraw ,deposite and moneyTransfer.**

**Create menu drive program for bank operations..**

package Assigment;

public class Account {

int accid;

String name,email,city;

long mobileno;

double balance;

public Account(int accid, String name, String email, String city, long mobileno, double balance) {

this.accid = accid;

this.name = name;

this.email = email;

this.city = city;

this.mobileno = mobileno;

this.balance = balance;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public long getMobileno() {

return mobileno;

}

public void setMobileno(long mobileno) {

this.mobileno = mobileno;

}

public int getAccid() {

return accid;

}

public double getAmount() {

return balance;

}

public double deposite(double amount)

{

System.out.println("----Deposite Amount----");

balance=balance+amount;

System.out.println("Balance after Deposite "+balance);

return this.balance;

}

public void widhtraw(double amount)

{

System.out.println("----Withdraw Amount-----");

if(balance>amount)

{

balance=balance - amount;

System.out.println("Balance atfer deducted amount "+balance);

}

else

{

System.out.println("Insufficient Funds");

}

}

public void moneytransfer(Account receiver,double amount )

{

System.out.println("----MoneyTrensfer---");

if(balance>amount)

{

this.balance=this.balance-amount;

receiver.balance=receiver.balance+amount;

System.out.println("Sender's balance after Money Transfer"+this.balance);

// System.out.println("Sender's balance after Money Transfer"+recevier.getbalance);

}

}

}

**AxisBank:**

package Day4\_1Assignment;

import Day4Assignment.Account;

import java.util.Scanner;

public class AxisBank {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("1.Create Account");

System.out.println("2.Display account");

System.out.println("3.Deposite Amount");

System.out.println("4.Withdraw Amount");

System.out.println("5.Money Transfer");

System.out.println("6.Exit");

Scanner sc=new Scanner(System.in);

Account a[]=new Account [10];

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

{

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

}

int choice;

int index=0;

do

{

System.out.println("Enter choice");

choice=sc.nextInt();

switch(choice)

{

case 1:

if(index<a.length)

{

System.out.println("Enter account details");

a[index]=new Account(sc.next(),sc.next(),sc.next(),sc.nextDouble());

index++;

}

else

{

System.out.println("array is full");

}

break;

case 2:

System.out.println("Display account details");

for(int i=0;i<index;i++)

{

if(a[i]!=null)

{

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

}else

{

System.out.println("array is full");

}

}

break;

case 3:

int act;

act=sc.nextInt();

for(int i=0;i<index;i++)

{

//if(a[i]!=null)

System.out.println("Deposite amount");

if(a[i].accid==act)

{

a[i].deposite(sc.nextDouble());

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

}

}

break;

case 4:

act=sc.nextInt();

for(int i=0;i<index;i++)

{

System.out.println("Withdraw amount");

if(a[i].accid==act)

{

a[i].withdraw(sc.nextDouble());

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

}

}

break;

case 5:

int act2;

act=sc.nextInt();

act2=sc.nextInt();

for(int i=0;i<index;i++)

{

if(a[i].accid==act)

{

for(int j=0;j<index;j++)

{

if(a[j].getAccid()==act2)

{

a[i].moneytransfer(a[j],sc.nextDouble());

System.out.println("Money Deducted on Sender account");

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

System.out.println("Money Received on benefishary account");

System.out.println(a[j]);

}

}

}

}

break;

default:

System.out.println("Invalid choice");

}

}while(choice!=6);

sc.close();

}

}

**Output:**

1.Create Account

2.Display account

3.Deposite Amount

4.Withdraw Amount

5.Money Transfer

6.Exit

Enter choice

1

Enter account details

Shubham

shubh@gmsil.com

Pune

45000

Enter choice

1

Enter account details

Suyash

suyash@gmail.com

Sangli

35000

Enter choice

2

Display account details

Account [accid=11, name=Shubham, email=shubh@gmsil.com, city=Pune, balance=45000.0]

Account [accid=12, name=Suyash, email=suyash@gmail.com, city=Sangli, balance=35000.0]

Enter choice

3

11

Deposite amount

10000

----Deposite Amount----

Balance after Deposite 55000.0

Account [accid=11, name=Shubham, email=shubh@gmsil.com, city=Pune, balance=55000.0]

Deposite amount

Enter choice

4

12

Withdraw amount

Withdraw amount

2000

----Withdraw Amount-----

Balance atfer deducted amount 33000.0

Account [accid=12, name=Suyash, email=suyash@gmail.com, city=Sangli, balance=33000.0]

Enter choice

5

11

12

10000

----MoneyTrensfer---

Sender's balance after Money Transfer45000.0

Money Deducted on Sender account

Account [accid=11, name=Shubham, email=shubh@gmsil.com, city=Pune, balance=45000.0]

Money Received on benefishary account

Account [accid=12, name=Suyash, email=suyash@gmail.com, city=Sangli, balance=43000.0]

Enter choice

6

Invalid choice

**8:Write a program to create student class with data members rollno, marks1,mark2,mark3.**

**Accept data (acceptInfo()) and display using display member function.**

**Also display total,percentage and grade.**

**Student:**

package Classes;

public class Student {

private int rollno,marks1,marks2,marks3,total;

private double per;

private String grade;

public static int counter=100;

public Student(int marks1,int marks2,int marks3)

{

counter++;

this.rollno=counter;

this.marks1=marks1;

this.marks2=marks2;

this.marks3=marks3;

}

private void calTotal()

{

total=marks1+marks2+marks3;

System.out.println("Total marks="+total);

}

private void CalPercentage()

{

per=total/3;

System.out.println("Percentage="+per);

}

private void grade() {

if(per>90 && per<=100)

{

grade="A+";

}

else if(per>80 && per<=90)

{

grade="A";

}

else if(per>70 && per<=80)

{

grade="B+";

}

else if(per>60 && per<=70)

{

grade="B";

}

else if(per>50 && per<=60)

{

grade="C+";

}

else if(per>50 && per<=60)

{

grade="C";

}

else if(per>=40 && per<=50)

{

grade="D";

}

else if(per>0 && per<40)

{

grade="Fail";

}

System.out.println("Grade="+grade);

}

public void display()

{

System.out.println("Roll no="+rollno);

calTotal();

CalPercentage();

grade();

}

}

**StudentTester:**

package Tester;

import Classes.Student;

import java.util.Scanner;

public class StudentTester {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc=new Scanner(System.in);

Student s[]=new Student [5];

//int index=0;

System.out.println("Enter 5 student marks");

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

{

s[i]=new Student(sc.nextInt(),sc.nextInt(),sc.nextInt());

if(s[i]!=null)

{

s[i].display();

}

}

sc.close();

}

}

**Output:**

Enter 5 student marks

45

67

89

Roll no=1

Total marks=201

Percentage=67.0

Grade=B

34

67

98

Roll no=2

Total marks=199

Percentage=66.0

Grade=B

99

98

97

Roll no=3

Total marks=294

Percentage=98.0

Grade=A+

76

67

56

Roll no=4

Total marks=199

Percentage=66.0

Grade=B

45

98

76

Roll no=5

Total marks=219

Percentage=73.0

Grade=B+

**9:Create Student class with rollno,name address.**

**Write business logic for auto incrment of rollno(don't accept roll no from user)**

**Write parameterised constr for accepting name and address only**

**Write getter and setter and display function**

**9.1 Test Student class by creating 5 diff object.and display aal details(chk rollno created automatically)**

**9.2 Create an array of 5 students and show only names**

**9:Create diff package and add class inside that.**

**Try to access one package class in another package....**

**(chk default access specifier)**

**Student:**

public class Student {

private int rollno;

private String name,address;

static int counter=1000;

public Student(String name,String Address)

{

counter++;

this.rollno=counter;

this.name=name;

this.address=Address;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public void display()

{

System.out.println("Rollno="+rollno+" Name="+name+" Address="+address);

}

}

**StudentTester:**

package Tester;

import java.util.Scanner;

import Classes.Student;

public class StudentTester {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc=new Scanner(System.in);

Student [] s= new Student[5];

int i;

System.out.println(" enter the Name of Student And Address");

for(i=0;i<s.length;i++)

{

s[i]=new Student(sc.next(),sc.next());

System.out.println("Name of Student="+s[i].getName());

// s[i].display();

}

sc.close();

}

}

**Output:**

enter the Name of Student And Address

Suyash

Sangli

Name of Student=Suyash

Shubham

Pune

Name of Student=Shubham

Ranjeet

Kolhapur

Name of Student=Ranjeet

Ruturaj

Kolhapur

Name of Student=Ruturaj

Abhishek

Pune

Name of Student=Abhishek

**Task 1**

**Create a class Point2D , under package "com.cdac.geometry" for representing a point in x-y co-ordinate system.**

**1.1 Create a parameterized constructor to accept x & y co-ords.**

**1.2 Add public String show()) --to return point's x & y co-ords**

**1.3 Add isEqual method to Point2D class : boolean returning method : must return true if both points are having same x,y co-ords or false otherwise.**

**1.4 Add a method , calculateDistance , to calc distance between 2 points**

**Hint : use distance formula.**

**1.5 Create a driver class(for UI) , in the "tester" package "TestPoints" , with main(..)**

**Ask user , how many points to plot? :**

**Create suitable array.**

**1.6 Accept x,y co-ordinates for all the points n store it suitably.**

**1.7 Display x,y co-ordinates of all the points plotted so far ,using for-each loop.**

**1.8 Accept 2 indices from user .**

**Find out if the points at these indices are same or different (Hint : isEqual)**

**Print the message accordingly.**

**If points are not same , display distance between these 2 points.**

**Point:**

package com;

public class Point {

private int x,y;

private double distance;

public Point(int x,int y)

{

this.x=x;

this.y=y;

}

public String show()

{

return "X-co-ordinate="+x+" Y-co-ordinate="+y;

}

public Boolean isEqual(Point p2)

{

if((this.x==p2.x) && (this.y==p2.y))

{

return true;

}

else

{

return false;

}

}

public void caldistance(Point p2)

{

distance=Math.sqrt( (((this.x-p2.x)\*(this.x-p2.x))+((this.y-p2.y)\*(this.y-p2.y))));

System.out.println("Distance between two points is "+distance);

}

@Override

public String toString()

{

return "X-co-ordinate="+x+" Y-co-ordinate"+y+" Distance between two points is"+distance;

}

}

**PointTester:**

package Day4Assignment;

public class PointTester {

public static void main(String[] args) {

// TODO Auto-generated method stub

Scanner sc=new Scanner(System.in);

int n;

System.out.println("Enter the no. of points which are to plot");

n=sc.nextInt();

int choice;

int index=0;

int i;

Point p []=new Point [n];

for(i=0;i<p.length;i++)

{

p[i]=new Point(0,0);

}

do

{

System.out.println("Enter choice");

choice=sc.nextInt();

switch(choice)

{

case 1:

System.out.println("Enter the points to plot");

if(index<p.length)

{

p[index]=new Point(sc.nextInt(),sc.nextInt());

index++;

}

break;

case 2:

Point p2;

System.out.println("Enter the co-ordinates of point");

p2=new Point(sc.nextInt(),sc.nextInt());

for(int k=0;k<index;k++)

{

if(p[k].isEqual(p2)==true)

{

System.out.println("Points Are Same");

}

else

{

p[k].caldistance(p2);

}

}

break;

default:

System.out.println("Invalid choice");

break;

}

}while(choice!=3);

sc.close();

}

}

**Output:**

Enter the no. of points which are to plot

4

Enter choice

1

Enter the points to plot

2

3

Enter choice

1

Enter the points to plot

3

4

Enter choice

1

Enter the points to plot

4

5

Enter choice

5

Invalid choice

Enter choice

1

Enter the points to plot

5

6

Enter choice

2

Enter the co-ordinates of point

2

3

Points Are Same

Distance between two points is 1.4142135623730951

Distance between two points is 2.8284271247461903

Distance between two points is 4.242640687119285

Enter choice

3

Invalid choice

**Task 2**

**2.1 Can you arrange Fruit,Apple,Orange,Mango in inheritance hierarchy ?**

**Use tight encapsulation**

**.**

**2.2 Properties (instance variables) : color : String , weight : double , name:String, fresh : boolean**

**2.3 Add suitable constructor.**

**2.4 Override toString correctly to return state of all fruits (return only : name ,color , weight )**

**2.5 Add a taste() method to return String form of the taste of the Fruit**

**eg : public String taste()**

**For Fruit : Can you identify taste of any general fruit ?**

**So return : "no specific taste"**

**Apple : should return "sweet n sour"**

**Mango : should return "sweet"**

**Orange : should return "sour"**

**2.6 Add specific functionality , in the sub classes**

**In Mango : public void pulp() {Display name n color of the fruit + a mesg creating pulp!}**

**In Orange : public void juice() {Display name n weight of the fruit + a mesg extracting juice!}**

**In Apple : public void jam() {Display name of the fruit + a mesg making jam!}**

**2.7 Add all of above classes under the package "com.app.fruits"**

**2.8 Create java application FruitBasket , with main method , as a tester**

**2.9 Prompt user for the basket size n create suitable data structure**

**2.10 Supply options**

**1. Add Mango**

**2. Add Orange**

**3. Add Apple**

**NOTE : You will be adding a fresh fruit in the basket , in all of above options.**

**4. Display names of all fruits in the basket.**

**5. Display name,color,weight , taste of all fresh fruits , in the basket.**

**6. Mark a fruit in a basket , as stale(=not fresh)**

**i/p : index**

**o/p : error message (in case of invalid index) or mark it stale**

**7. Mark all sour fruits stale**

**Hint : Use equals() method of the String class.**

**8. Invoke fruit specific functionality (pulp / juice / jam)**

**i/p : index**

**Invoke correct functionality (pulp / juice / jam)**

**10. Exit**

**Fruit:**

package Day4Assignment;

public class Fruit {

private String color;

private double weight;

protected String name;

protected String state;

public Fruit(String color,double weight,String name,String state)

{

this.color=color;

this.weight=weight;

this.name=name;

this.state=state;

}

public String taste()

{

return "No specific Taste";

}

public Boolean isfresh()

{

if(state=="NotFresh")

{

return true;

}

else

{

return false;

}

}

@Override

public String toString()

{

return "Name="+name+" Color="+color+" Weight="+weight+" State="+state;

}

}

**Mango:**

package Day4Assignment;

public class Mango extends Fruit{

public Mango(String color, double weight, String name, String fresh) {

super(color, weight, name, fresh);

}

@Override

public String taste()

{

return "Sweet";

}

public void pulp()

{

System.out.println("Mango for creating pulp");

}

}

package Day4Assignment;

public class Orange extends Fruit{

public Orange(String color, double weight, String name, String fresh) {

super(color, weight, name, fresh);

}

@Override

public String taste()

{

return "Sour";

}

public void juice()

{

System.out.println("Orange for extracting juice");

}

}

**Apple:**

package Day4Assignment;

public class Apple extends Fruit {

//private String jam;

public Apple(String color, double weight, String name, String fresh) {

super(color, weight, name, fresh);

}

@Override

public String taste()

{

return "Sweet and Sour";

}

public void jam()

{

System.out.println("Apple for making jam");

}

}

**FruitTester:**

package Day4Assignment;

import java.util.Scanner;

public class FruitTester {

public static void main(String[] args) {

// TODO Auto-generated method stub

int n;

Scanner sc=new Scanner(System.in);

System.out.println("Enter no. of fruits");

n=sc.nextInt();

Fruit f []=new Fruit[n];

Fruit arr []=new Fruit[n];

Fruit a []=new Fruit [n];

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

{

f[i]=new Fruit(null,0,null,null);

}

int choice;

int index=0;

int j=0;

int k=0;

do {

System.out.println("Enter choice");

choice=sc.nextInt();

switch(choice)

{

case 1:

if(index<f.length)

{

System.out.println("Add mango");

f[index]=new Mango("Yellow",5,"Mango","Fresh");

index++;

}

else

{

System.out.println("Array is full");

}

break;

case 2:

if(index<f.length)

{

System.out.println("Add Orange");

f[index]=new Orange("Orange",4,"Orange","Fresh");

index++;

}

else

{

System.out.println("Array is full");

}

break;

case 3:

if(index<f.length)

{

System.out.println("Add Apple");

f[index]=new Apple("Apple",6,"Red","NotFresh");

index++;

}

else

{

System.out.println("Array is full");

}

break;

case 4:

System.out.println("Display name all the fruits");

for(int i=0;i<index;i++)

{

if(f[i]!=null)

{

System.out.println(f[i].name);

}

}

break;

case 5:

System.out.println("Fruits which are fresh");

for(int i=0;i<index;i++)

{

if(f[i].state=="Fresh")

{

if(f[i]!=null)

{

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

}

}

}

break;

case 6:

System.out.println("Fruits which are not fresh");

for(int i=0;i<index;i++)

{

if(f[i].isfresh()==true)

{

if(f[i]!=null)

{

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

}

}

}

break;

case 7:

System.out.println("All sour fruits are stale");

for(int i=0;i<index;i++)

{

if(f[i]!=null)

{

if(f[i].taste()=="Sour")

{

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

}

}

}

break;

case 8:

for(int i=0;i<index;i++)

{

if(f[i] instanceof Mango)

{

((Mango)f[i]).pulp();

}

if(f[i] instanceof Apple)

{

((Apple)f[i]).jam();

}

if(f[i] instanceof Orange)

{

((Orange)f[i]).juice();

}

}

break;

default:

System.out.println("Invalid choice");

}

}while(choice!=9);

sc.close();

}

}

**Output:**

Enter no. of fruits

6

Enter choice

1

Add mango

Enter choice

1

Add mango

Enter choice

2

Add Orange

Enter choice

2

Add Orange

Enter choice

3

Add Apple

Enter choice

3

Add Apple

Enter choice

4

Display name all the fruits

Mango

Mango

Orange

Orange

Red

Red

Enter choice

5

Fruits which are fresh

Name=Mango Color=Yellow Weight=5.0 State=Fresh

Name=Mango Color=Yellow Weight=5.0 State=Fresh

Name=Orange Color=Orange Weight=4.0 State=Fresh

Name=Orange Color=Orange Weight=4.0 State=Fresh

Enter choice

6

Fruits which are not fresh

Name=Red Color=Apple Weight=6.0 State=NotFresh

Name=Red Color=Apple Weight=6.0 State=NotFresh

Enter choice

7

All sour fruits are stale

Name=Orange Color=Orange Weight=4.0 State=Fresh

Name=Orange Color=Orange Weight=4.0 State=Fresh

Enter choice

8

Mango for creating pulp

Mango for creating pulp

Orange for extracting juice

Orange for extracting juice

Apple for making jam

Apple for making jam

Invalid choice

Enter choice

**3. Solve this.**

**Fresh business scenario to apply inheritance , polymorphism n abstraction to emp based organization scenario.**

**Create Emp based organization structure --- Emp , Mgr , Worker**

**All of above classes must be in --com.app.org**

**3.1 Emp state--- id(int), firstName, lastName , deptId , basic(double)**

**emp id MUST be automatically generated.**

**Behaviour ---1. get emp details -- using toString.**

**2. compute net salary**

**3.2 Mgr state ---id,name,basic,deptId , perfBonus**

**Behaviour ----1. get mgr details : using toString.**

**2. compute net salary (formula: basic+perfBonus) -- override computeNetSalary**

**3. get performance bonus. --add a new method to return bonus.(getter)**

**3.3 Worker state --id,name,basic,deptId,hoursWorked,hourlyRate**

**Behaviour---**

**1. get worker details -- : override toString.**

**2. compute net salary (formula: = basic+(hoursWorked\*hourlyRate) --override computeNetSalary**

**3. get hrlyRate of the worker -- add a new method to return hourly rate of a worker.(getter)**

**Organize classes in inheritance hierarchy.**

**NOTE : toString method SHOULD NOT include the net salary of the employee**

**Write TestOrganization in "tester" package.**

**Create suitable array to store organization details.**

**Provide following options**

**1. Hire Manager**

**i/p : manager details , except id**

**2. Hire Worker**

**i/p : worker details , except id**

**3. Display information of all employees(toString) including net salary(computeNetSalary) using single for-each loop.**

**4. Update basic salary**

**i/p : emp id , salary increment**

**In case of invalid emp id , either : display error message directly**

**OR (still better)**

**throw custom exception n handle it in centralized manner**

**(This can be added later!)**

**10 Exit**

**Employee:**

package Day4Assignment;

public class Employee {

protected int empid;

protected String firstname;

protected String lastname;

protected int deptid;

protected double basicsal;

public static int counterid=0;

public Employee(String firstname, String lastname, int deptid, double basicsal) {

counterid++;

this.empid = counterid;

this.firstname = firstname;

this.lastname = lastname;

this.deptid = deptid;

this.basicsal = basicsal;

}

public double getBasicsal() {

return basicsal;

}

public void setBasicsal(double basicsal) {

this.basicsal =this.basicsal+ basicsal;

}

public double netsal()

{

return basicsal;

}

@Override

public String toString() {

return "Employee [empid=" + empid + ", firstname=" + firstname + ", lastname=" + lastname + ", deptid=" + deptid

+ ", basicsal=" + netsal() + "]";

}

}

package Day4Assignment;

public class Manager extends Employee{

protected double perbonus;

public Manager(String firstname, String lastname, int deptid, double basicsal, double perbonus) {

super(firstname, lastname, deptid, basicsal);

this.perbonus = perbonus;

}

@Override

public double netsal()

{

return this.basicsal+perbonus;

}

public double getBasicsal() {

return basicsal;

}

public void setBasicsal(double basicsal) {

this.basicsal =this.basicsal+ basicsal;

}

@Override

public String toString() {

return "Manager [perbonus=" + perbonus + ", empid=" + empid + ", firstname=" + firstname + ", lastname="

+ lastname + ", deptid=" + deptid + ", basicsal=" + basicsal +",Netsal="+netsal()+ "]";

}

public double getPerbonus() {

return perbonus;

}

}

**Worker:**

package Day4Assignment;

public class Worker extends Employee{

private double hoursworked,hourlyrate;

public Worker(String firstname, String lastname, int deptid, double basicsal,double hoursworked,double hourlyrate)

{

super(firstname,lastname,deptid,basicsal);

this.hoursworked=hoursworked;

this.hourlyrate=hourlyrate;

}

@Override

public double netsal()

{

return this.basicsal +(hoursworked\*hourlyrate);

}

@Override

public String toString() {

super.toString();

return "Worker [hoursworked=" + hoursworked + ", hourlyrate=" + hourlyrate +" Netsalary="+netsal()+ "]";

}

public double getHourlyrate() {

return hourlyrate;

}

}

**EmployeeTester:**

package Day4Assignment;

import java.util.Scanner;

public class EmployeeTester {

public static void main(String[] args) {

// TODO Auto-generated method stub

System.out.println("1.Hire Manager");

System.out.println("2.Hire Worker");

System.out.println("3.Display Employee Details");

System.out.println("4.Update Net Salary");

System.out.println("5.Exit");

Scanner sc=new Scanner(System.in);

int n;

System.out.println("Enter no. of employees");

n=sc.nextInt();

Employee e[]=new Employee[n];

int choice;

int index=0;

do {

System.out.println("Enter choice");

choice=sc.nextInt();

switch(choice)

{

case 1:

System.out.println("Hire Manager");

if(index<e.length)

{

e[index]=new Manager(sc.next(),sc.next(),sc.nextInt(),sc.nextDouble(),sc.nextDouble());

index++;

}

break;

case 2:

System.out.println("Hire Worker");

if(index<e.length)

{

e[index]=new Worker(sc.next(),sc.next(),sc.nextInt(),sc.nextDouble(),sc.nextDouble(),sc.nextDouble());

index++;

}

break;

case 3:

System.out.println("Display employee details");

for(Employee E:e)

{

if(E!=null)

{

System.out.println(E);

}

}

break;

case 4:

int id;

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

id=sc.nextInt();

for(int i=0;i<index;i++)

{

if(e[i]!=null) {

if(e[i].empid==id)

{

System.out.println("Update Salary");

if(e[i] instanceof Manager)

{

((Manager)e[i]).setBasicsal(sc.nextDouble());

System.out.println("Updated Salary="+((Manager)e[i]).getBasicsal());

}

if(e[i] instanceof Worker)

{

((Worker)e[i]).setBasicsal(sc.nextDouble());

System.out.println("Updated Salary="+((Worker)e[i]).getBasicsal());

}

}

}

}

break;

default:

System.out.println("Invlid choice");

break;

}

}while(choice!=5);

sc.close();

}

}

**Output:**

1.Hire Manager

2.Hire Worker

3.Display Employee Details

4.Update Net Salary

5.Exit

Enter no. of employees

5

Enter choice

1

Hire Manager

Suyash

Gavade

2

45000

10000

Enter choice

2

Hire Worker

Adesh

Patil

3

35000

8

500

Enter choice

3

Display employee details

Manager [perbonus=10000.0, empid=1, firstname=Suyash, lastname=Gavade, deptid=2, basicsal=45000.0,Netsal=55000.0]

Worker [hoursworked=8.0, hourlyrate=500.0, empid=2, firstname=Adesh, lastname=Patil, deptid=3, basicsal=35000.0]

Enter choice

4

Enter id

1

Update Salary

5000

Updated Salary=50000.0

Enter choice

4

Enter id

2

Update Salary

5000

Updated Salary=40000.0

Enter choice

**1. Create abstract class Shape --state : x,y**

**Abstract Method --public double area();**

**public String toString() : to ret x & y**

**Why will area() be abstract in Shape class ?????????**

**2. Circle -- x,y,radius**

**Concrete overriding Method --public double area() : ret area of circle**

**public String toString() : ret x, y & radius**

**3. Rectangle -- x,y,w,h**

**Concrete overriding Method --public double area() : ret area of rectangle**

**public String toString() : ret x, y , width & height**

**4. Square-- x,y,side**

**Concrete overriding Method --public double area() : ret area of square**

**public String toString() : ret x, y , side**

**5. Create a ShapeFactory class**

**Add a static method(generateShape) to return randomly generated shape.**

**Hint : random no generator**

**6. Create a Tester . Invoke ShapeFactory's generateShape() method , in a for-loop (5 times)**

**to display details & area of each shape**

**Shapes:**

public abstract class Shapes {

protected String x;

protected String y;

public abstract double area();

@Override

public String toString()

{

return x + y;

}

}

**Circle:**

package Day4Assignment;

public class Circle extends Shapes {

private int radius;

public double area() {

double area = 3.14 \* radius \* radius;

return area;

}

public void setRadius(int radius) {

this.radius = radius;

}

}

**Rectangle:**

package Day4Assignment;

public class Rectangle extends Shapes {

private int w;

private int h;

public double area() {

double area = w \* h;

return area;

}

public void setW(int w) {

this.w = w;

}

public void setH(int h) {

this.h = h;

}

}

**Square:**

package Day4Assignment;

public class Square extends Shapes {

private int side;

public double area() {

double area = side \* side;

return area;

}

public void setSide(int side) {

this.side = side;

}

}

**ShapeFactory:**

package Day4Assignment;

import java.util.Random;

public class ShapeFactory {

public static int generateShape() {

Random random = new Random();

int upper\_bound = 4;

int num = random.nextInt(upper\_bound);

return num;

}

}

**Area Tester:**

package Day4Assignment;

import java.util.Scanner;

public class AreaTester {

public static void main(String[] args) {

Scanner in = new Scanner(System.in);

int run = 0;

while (run != 5) {

int num = ShapeFactory.generateShape();

switch (num) {

case 1:

System.out.println("\nShape generated: Circle");

Circle circle = new Circle();

System.out.println("Enter the radius of the Circle: ");

circle.setRadius(in.nextInt());

System.out.println("Area of your circle is: " + circle.area());

run++;

break;

case 2:

System.out.println("\nShape generate: Rectangle");

Rectangle rectangle = new Rectangle();

System.out.println("Enter the Width of the Rectangle: ");

rectangle.setW(in.nextInt());

System.out.println("Enter the Height of the Rectangle: ");

rectangle.setH(in.nextInt());

System.out.println("Area of your Rectangle is: " + rectangle.area());

run++;

break;

case 3:

System.out.println("\nShape generated: Square");

Square square = new Square();

System.out.println("Enter the length of the side of the Square: ");

square.setSide(in.nextInt());

System.out.println("Area of the Square is: " + square.area());

run++;

break;

default:

System.out.println("\nShape not formed. Please try again!");

run++;

break;

}

}

System.out.println("Looop Endss...!!");

in.close();

}

}

**Output:**

Shape generate: Rectangle

Enter the Width of the Rectangle:

3

Enter the Height of the Rectangle:

4

Area of your Rectangle is: 12.0

Shape generated: Circle

Enter the radius of the Circle:

2

Area of your circle is: 12.56

Shape generated: Square

Enter the length of the side of the Square:

4

Area of the Square is: 16.0

Shape generated: Circle

Enter the radius of the Circle:

4

Area of your circle is: 50.24

Shape generated: Square

Enter the length of the side of the Square:

5

Area of the Square is: 25.0

Looop Endss...!!

**2.** **Define an interface and implement it in any class wherever it is required.**

**Pre-condition: Employee, Date and Shape class should be created.**

**Problem Statement:**

**Define an interface Printable with a method print(). Implement this interface in Employee, Shape and Date class.**

package Day4\_1Assignment;

public interface IPrintable {

void print();

}

**Employee:**

package Day4\_1Assignment;

public class Employee implements IPrintable {

private int empid;

private String name;

private double salary;

public Employee() {

this.empid = 000;

this.name = "Unknown";

this.salary = 0.000;

}

public Employee(int empid, String name, double salary) {

this.empid = empid;

this.name = name;

this.salary = salary;

}

@Override

public String toString() {

return "Employee Details \nID:" + this.empid + " Name:" + this.name + " Salary:" + this.salary;

}

@Override

public void print() {

System.out.println("Enter Employee ID, Name, Salary");

}

}

**Date:**

package Day4\_1Assignment;

public class Date implements IPrintable {

private int dd, mm, yy;

public Date() {

this.dd = 1;

this.mm = 6;

this.yy = 1960;

}

public Date(int dd, int mm, int yy) {

this.dd = dd;

this.mm = mm;

this.yy = yy;

}

@Override

public String toString() {

return "Date: " + dd + "/" + mm + "/" + yy;

}

@Override

public void print() {

System.out.println("Enter Date in dd/mm/yyyy format");

}

}

**Shape:**

package Day4\_1Assignment;

public abstract class Shape implements IPrintable {

public abstract double area();

@Override

public void print() {

}

}

Rectangle:

package Day4\_1Assignment;

public class Rectangle extends Shape {

private double n1, n2;

public Rectangle() {

this.n1 = 1.00;

this.n2 = 1.00;

}

public Rectangle(double n1, double n2) {

this.n1 = n1;

this.n2 = n2;

}

@Override

public double area() {

return n1 \* n2;

}

public void print() {

System.out.println("Enter values of x and y to calculate area of rectangle:");

}

}

**Circle:**

package Day4\_1Assignment;

public class Circle extends Shape {

private double radius;

public Circle() {

radius = 1.0;

}

public Circle(double radius) {

this.radius = radius;

}

@Override

public double area() {

return 3.14 \* radius \* radius;

}

public void print() {

System.out.println("Enter the value of radius to calculate area of circle:");

}

}

**Tester:**

package Day4\_1Assignment;

import java.util.Scanner;

public class Tester\_Interface {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// testing employee class

Employee emp = new Employee();

emp.print();

emp = new Employee(sc.nextInt(), sc.next(), sc.nextDouble());

System.out.println(emp);

System.out.println();

// testing date class

Date d1 = new Date();

d1.print();

d1 = new Date(sc.nextInt(), sc.nextInt(), sc.nextInt());

System.out.println(d1);

System.out.println();

// testing shape class

System.out.println("What to calculate ? \n1.Area of Rectangle \n2.Area of Circle \n3.Exit");

int choice;

do {

System.out.println("Enter choice: ");

choice = sc.nextInt();

switch (choice) {

case 1:

Shape sh = new Rectangle();

sh.print();

sh = new Rectangle(sc.nextDouble(), sc.nextDouble());

double d = sh.area();

System.out.println("Area Of Rectangle: " + d);

break;

case 2:

Shape sh1 = new Circle();

sh1.print();

sh1 = new Circle(sc.nextDouble());

double d2 = sh1.area();

System.out.println("Area of Circle:" + d2);

break;

case 3:

System.out.println("Exiting the code.... ");

break;

default:

System.out.println("Invalid Input!!!");

break;

}// switch

} while (choice <= 2);

sc.close();

}

}

**Output:**

Enter Employee ID, Name, Salary

101 Ranjeet 26000

Employee Details

ID:101 Name:Ranjeet Salary:26000.0

Enter Date in dd/mm/yyyy format

24 3 2023

Date: 24/3/2023

What to calculate ?

1.Area of Rectangle

2.Area of Circle

3.Exit

Enter choice:

1

Enter values of x and y to calculate area of rectangle:

4 5

Area Of Rectangle: 20.0

Enter choice:

2

Enter the value of radius to calculate area of circle:

4

Area of Circle:50.24

Enter choice:

3

Exiting the code....