

Concept of Programming Assignments

Name: Mohan Malappa Talavar
Roll No: 233169

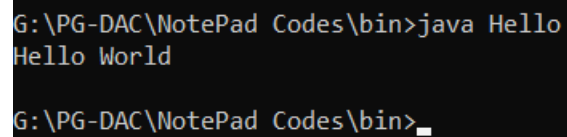
PRN: 230341220099
Course: PG-DAC | B Batch

Assignment No: 1

Q1: Write program to test Hello World.

```
public class Q1_Hello  
  
    public static void main(String[] args) {  
  
        System.out.println("Hello World");  
  
    }
```

OUTPUT:



```
G:\PG-DAC\NotePad Codes\bin>java Hello  
Hello World  
G:\PG-DAC\NotePad Codes\bin>_
```

Q2 Write a program to addition of two numbers

```
import java.util.Scanner;

public class Q2_Add2N
{
    public static void main(String[] args)
    {
        int a, b,c;

        System.out.println("Enter two numbers= ");

        Scanner sc= new Scanner(System.in);

        a= sc.nextInt();

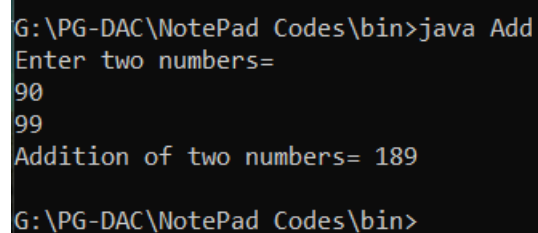
        b= sc.nextInt();

        c= a+b;

        System.out.println("Addition of two numbers= "+c);

    }
}
```

OUTPUT:



```
G:\PG-DAC\NotePad Codes\bin>java Add
Enter two numbers=
90
99
Addition of two numbers= 189
G:\PG-DAC\NotePad Codes\bin>
```

Q3:Write a program to swap two numbers.

```
import java.util.Scanner;

public class Q3_Swap
{
    public static void main(String[] args)
    {
        int a,b,c;

        System.out.println("Enter two Numbers= ");

        Scanner sc=new Scanner(System.in);

        a= sc.nextInt();

        b= sc.nextInt();

        c=a;

        a=b;

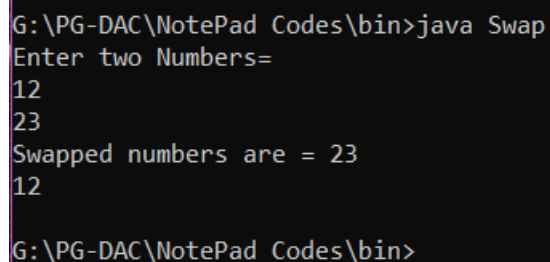
        b=c;

        System.out.println("Swapped numbers are = "+a);

        System.out.println(b);

    }
}
```

OUTPUT:



```
G:\PG-DAC\NotePad Codes\bin>java Swap
Enter two Numbers=
12
23
Swapped numbers are = 23
12
G:\PG-DAC\NotePad Codes\bin>
```

Q4 Write a program to find factorial of a given number.

```
import java.util.Scanner;

public class Q4_Factorial {

    public static void main(String[] args) {

        int n , fact=1;

        System.out.println("Enter a Number");

        Scanner sc = new Scanner(System.in);

        n = sc.nextInt();

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

        {

            fact *=i;

        }

        System.out.println("The value of factorial "+fact);

        sc.close();

    }

}
```

Output:

Enter a Number

8

The value of factorial: 40320

Q5 Write a program to find m to the power n.

```
import java.util.Scanner;

public class Q5_Pow {

    public static void main(String[] args) {

        int x,n, pow = 1;

        System.out.println("Enter a Number ");
        Scanner sc = new Scanner(System.in);
        x = sc.nextInt();

        System.out.println("Enter a power ");
        Scanner sd = new Scanner(System.in);
        n = sd.nextInt();

        for (int i = 1; i <= n; i++) {
            pow *= x;
        }

        System.out.println("The value of " +x+ " power " +n+ "
is: " +pow);
        sc.close();
    }

}
```

Output:

```
Enter a Number
5
Enter a power
3
The value of 5 power 3 is: 125
```

Q6 Check if number is a prime number or not.

```
import java.util.Scanner;
public class Q6_Prime {

    public static void main(String[] args)
    {
        int n;
        Scanner sc=new Scanner (System.in);
        System.out.println("Enter a number to check Prime=");
        n= sc.nextInt();
        int k=0;

        for(int i=2; i<n; i++)
        {
            if(n%i==0)
                k++;
        }
        if(k>0)
            System.out.println(n + " is NOT a Prime Number");
        else
            System.out.println(n + " is a Prime Number");

        sc.close();
    } //main
}
```

Output:

```
Enter a number to check Prime=
8
8 is NOT a Prime Number
```

```
Enter a number to check Prime=
59
59 is a Prime Number
```

Q7 Sum of series :

// 1+2+3+....+n

```
import java.util.Scanner;
```

```
public class Q7_SumOFSeries {
```

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

```
        int n,sum = 0;
```

```
        System.out.println("Enter a Number");
```

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

```
        n = sc.nextInt();
```

```
        for(int i=0;i<=n; i++)
```

```
        {
```

```
            sum+= i;
```

```
        }
```

```
        System.out.println("The sum of numbers upto " +n+ " is: "+sum);
```

```
    }
```

```
}
```

Output

Enter a Number

8

The sum of numbers upto 8 is: 36

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

```
import java.util.Scanner;

public class SumOfEvenAndOddNumbers {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the value of n: ");
        int n = sc.nextInt();

        int evenSum = 0;
        int oddSum = 0;

        for (int i = 1; i <= n; i++) {
            if (i % 2 == 0) {
                evenSum += i;
            } else {
                oddSum += i;
            }
        }

        System.out.println("Sum of even numbers between 1 and " + n
+ " is " + evenSum);
        System.out.println("Sum of odd numbers between 1 and " + n +
" is " + oddSum);
    }
}
```

OUTPUT:

Enter the value of n:

12

Sum of even numbers between 1 and 12 is 42

Sum of odd numbers between 1 and 12 is 36

Q10 Write a program to enter a number and print its reverse.

```
import java.util.Scanner;
public class Q10_Rev {

    public static void main(String[] args) {

        int n,rem,rev=0;
        System.out.println("Enter a Number : ");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();

        while(n != 0)
        {
            rem = n%10;
            rev = rev*10 + rem;
            n /= 10;
        }
        System.out.println("Reverse Number is "+rev);
        sc.close();
    }
}
```

OUTPUT

```
Enter a Number :
123456789
Reverse Number is 987654321
```

Q11 Write a program to print all Prime numbers between 1 to n.

```
import java.util.Scanner;

public class Q11_AllPrime {

    public static void main(String[] args) {

        int n, prime;

        System.out.println("Enter a Positive Number: ");
        Scanner sc = new Scanner(System.in);
        n = sc.nextInt();
        System.out.println("Prime Numbers upto "+ n+ " are: ");

        for (int i = 2; i <= n; i++) {
            prime = 1;
            for (int j = 2; j < i; j++) {
                if (i % j == 0) {
                    prime = 0;
                    break;
                }
            }
            if (prime == 1)
                System.out.println(i);
        }
        sc.close();
    }
}
```

Output:

Prime Numbers upto 11 are:

2
3
5
7
11

Q12 Write a program to check entered number is Armstrong number or not.

```
import java.util.Scanner;

import java.lang.Math;

public class Q12_Armstrong {

    public static void main(String[] args) {

        int count = 0, num, sum = 0;

        System.out.println("Enter a number: ");

        Scanner sc = new Scanner(System.in);

        num = sc.nextInt();

        int temp = num;

        int temp1 = num;

        /// counting the digit

        do {

            temp = temp / 10;

            count++;

        } while (temp != 0);

        // assign

        int arr[] = new int[count];

        for (int i = 0; temp1 != 0; i++) {

            arr[i] = temp1 % 10;

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

            temp1 = temp1 / 10;

            // System.out.println(temp1);

        }

    }

}
```

```
// Armstrong logic
for (int i = 0; i < count; i++) {
    sum += Math.pow(arr[i], count);
}

if (sum == num)
    System.out.println("Armstrong Number");
else
    System.out.println("Not an Armstrong Number!!!");

sc.close();
}
}
```

OUTPUT

Enter a number:

371

| Armstrong Number

Enter a number:

456

Not an Armstrong Number!!!

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

```
import java.util.Scanner;

public class Q13_Greatest_3No {

    public static void main(String[] args) {

        int a,b,c;
        System.out.println("Enter the values of 3 no: ");
        Scanner sc=new Scanner(System.in);
        a=sc.nextInt();
        b=sc.nextInt();
        c=sc.nextInt();

        if(a>b)
        {
            if(a>c)
                System.out.println(a + " is Greatest");
        }
        else if(b>c)
        {
            if(b>a)
                System.out.println(b + " is Greatest");
        }
        else
            System.out.println(c + " is Greatest");

        sc.close();
    }
}
```

OUTPUT:

Enter the values of 3 no:

10

50

30

50 is Greatest

Q14 Create menu driven program for Pizza Shop. And display total amount.

```
import java.util.Scanner;
```

```
public class Q14_Pizza {
```

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

```
        int total = 0;
```

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

```
        int choice;
```

```
        System.out.println("Welcome to Pizza Shop. ");
```

```
        System.out.println("Our Menu ");
```

```
        System.out.println("1) Margerita Pizza Cost: 200Rs");
```

```
        System.out.println("2) Peri Peri Pizza Cost: 250Rs");
```

```
        System.out.println("3) Cheese Pizza Cost: 300Rs");
```

```
        System.out.println("4) Exit the cart");
```

```
        System.out.println("Add Items to cart!!");
```

```
        do {
```

```
            System.out.println("Enter your choices:");
```

```
            choice = sc.nextInt()
```

```
        switch (choice) {
```

```
        case 1:
```

```
            System.out.println("Enter the number of Margerita Pizza");
```

```
                int n1 = sc.nextInt();
```

```
                total += n1 * 200;
```

```
                System.out.println("Total Cost=" + total + "Rs");
```

```
                break;
```

case 2:

```
System.out.println("Enter the number of Peri Peri Pizza=");  
  
    int n2 = sc.nextInt();  
  
    total += n2 * 250;  
  
    System.out.println("Total Cost= " + total + "Rs");  
  
    break;
```

case 3:

```
System.out.println("Enter the number of Cheese Pizza");  
  
    int n3 = sc.nextInt();  
  
    total += n3 * 300;  
  
    System.out.println("Total Cost=" + total + "Rs");  
  
    break;
```

case 4:

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

default:

```
System.out.println("Invalid Input");  
  
    } // switch
```

```
} while (choice != 4);
```

```
System.out.println("THANK YOU! Visit again.. ");
```

```
sc.close();
```

```
}
```

OUTPUT:

Welcome to Pizza Shop.

Our Menu

1) Margerita Pizza Cost: 200Rs

2) Peri Peri Pizza Cost: 250Rs

3) Cheese Pizza Cost: 300Rs

4) Exit the cart

Add Items to cart!!

Enter your choices:

1

Enter the number of Margerita Pizza

1

Total Cost=200Rs

Enter your choices:

2

Enter the number of Peri Peri Pizza=

2

Total Cost= 700Rs

Enter your choices:

3

Enter the number of Cheese Pizza

3

Total Cost=1600Rs

Enter your choices:

4

Exiting the code..

THANK YOU! Visit again..

Q15 Create Menu driven program for array operations.

- 1:Read Array
- 2:Print Array
- 3:Search element in array
- 4:Reverse Array
- 5:Even number from array
- 6:sum of array element

```
import java.util.Scanner;
public class Q15_Array_Menu {

    public static void main(String[] args) {

        int size;
        System.out.println("Enter the size of array= ");
        Scanner sc=new Scanner(System.in);
        size= sc.nextInt();
        int arr[]= new int [size];

        System.out.println("1:Read Array");
        System.out.println("2:Print Array");
        System.out.println("3:Search element in array");
        System.out.println("4:Reverse Array");
        System.out.println("5:Even number from array");
        System.out.println("6:sum of array element");
        System.out.println("7:Exit..");
        int choice;
        do {
            System.out.println("Enter choice=");
            choice= sc.nextInt();

            switch(choice) {
                case 1:
                    System.out.println("Enter the elements
ofarray");

                    for(int i= 0;i<size; i++)
                    {
                        arr[i]= sc.nextInt();
                    }
                    break;
                case 2:
                    System.out.println("Elements of array=
");

                    for(int i= 0;i<size; i++)
                    {
                        System.out.println(arr[i]+ " ");
                    }
            }
        }
    }
}
```

```

        break;

    case 3:
        System.out.println("Enter the element to
search= ");

        int search= sc.nextInt();
        boolean isPresent= false;
        for(int i =0; i<size; i++)
        {
            if(search==arr[i])
            {isPresent= true;
            break;
            }
            else
                isPresent= false;
        }// for
        if(isPresent)
            System.out.println("Element
"+search+ " Found");
        else System.out.println(search+" Not
Found");

        break;

    case 4: //reverse array

        System.out.println("Reversed Array= ");
        for(int i=size-1; i>=0; i--)
        {
            System.out.println(arr[i]);
        }
        break;

    case 5: //even number from array;
        int k=0;
        System.out.println("Even numbers in given
array=");

        for(int i=0; i<size; i++)
        {
            if(arr[i]%2==0)
                {System.out.println(arr[i]+"
");
                k++;
            }
        }
        if(k==0)
            System.out.println("Even numbers
not found!");

        break;

    case 6:

```

```
        int sum= 0;
        for(int i=0; i<size; i++)
        {
            sum+= arr[i];
        }
        System.out.println("Sum of array= "+sum);
        break;

    case 7:
        System.out.println("Exiting the code");
        break;
    default:
        System.out.println("Invalid Input!");
        break;

} // switch

}while(choice!= 7);//do

    sc.close();
}

} // end of main
```

OUTPUT:

```
Enter the size of array=
4
1:Read Array
2:Print Array
3:Search element in array
4:Reverse Array
5:Even number from array
6:sum of array element
7:Exit..
Enter choice=
1
Enter the elements of array
11
22
33
44
Enter choice=
2
Elements of array=
11
22
33
44
Enter choice=
3
Enter the element to search=
44
Element 44 Found
Enter choice=
4
Reversed Array=
44
33
22
11
Enter choice=
5
Even numbers in given array=
22
44
Enter choice=
6
Sum of array= 110
Enter choice=
7
Exiting the code
```

Q16 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 Q16_Add2Array {

    public static void main(String[] args) {
        int a , b, c;
        System.out.println("Enter size of 1st array= ");
        Scanner sc= new Scanner(System.in);
        a = sc.nextInt();

        int arr1[]= new int[a];
        System.out.println("Enter the elements of 1st Array");
        for(int i=0; i<a; i++)
        {
            arr1[i]= sc.nextInt();
        }

        System.out.println("Enter size of 2nd array= ");
        b = sc.nextInt();

        int arr2[]= new int[b];
        System.out.println("Enter the elements of 2nd Array");
        for(int i=0; i<b; i++)
        {
            arr2[i]= sc.nextInt();
        }

        c= a+b;
        int arr3[]= new int[c];

        for(int i=0; i<a; i++)
        {
            arr3[i]= arr1[i];
        }

        for(int i=0; i<b; i++)
        {
            arr3[a+i]= arr2[i];
        }
    }
}
```

```
// Represent final array with "For Each loop"
System.out.println("Concatenation of 1st and 2nd array=
");
    for(int no: arr3)
    {
        System.out.println(no + " ");
    }

    sc.close();
}
}
```

OUTPUT:

```
Enter size of 1st array=
4
Enter the elements of 1st Array
11
22
33
44
Enter size of 2nd array=
3
Enter the elements of 2nd Array
55
66
77
Concatenation of 1st and 2nd array=
11
22
33
44
55
66
77
```

#ASSIGNMENT 2:-

1. 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.

Date Class :-

```
public class Student {

    private int rollno,marks1,marks2,marks3,total;
    String grade;
    double percentage;

    public Student()
    {
        System.out.println("---Default constructor---");
        rollno = 0;
        marks1 = 0;
        marks2 = 0;
        marks3 = 0;
    }

    public Student(int rollno, int marks1, int marks2, int marks3)
    {
        super();
        System.out.println("---Parameterized constructor---");
        this.rollno = rollno;
        this.marks1 = marks1;
        this.marks2 = marks2;
        this.marks3 = marks3;
    }

    public void acceptInfo(int rollno,int marks1,int marks2,int
marks3)
    {
        System.out.println("---Inside AcceptInfo---");
        this.rollno = rollno;
        this.marks1 = marks1;
        this.marks2 = marks2;
        this.marks3 = marks3;
    }
}
```

```
public void total()
{
    total = (marks1+marks2+marks3);
}

public void percentage()
{
    percentage = total/3;
}

public void grade()
{
    if (percentage>=80.0)
    {
        grade = "A+";
    }
    else if (percentage<80.0 && percentage >=70.0)
    {
        grade = "A";
    }
    else if (percentage<70.0 && percentage>=60.0)
    {
        grade = "B";
    }
    else if (percentage<60.0 && percentage>=50.0)
    {
        grade = "C";
    }
    else
    {
        grade = "F";
    }
}

public void display()
{
    total();
    percentage();
    grade();
    System.out.println("Roll No: "+rollno+" Percentage:
    "+percentage+" Grade: "+grade);
}

}
```


TesterDate:-

```
import java.util.Scanner;

public class TesterStudent {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        //default constructor
        Student s1 = new Student();
        s1.display();
        System.out.println("-----");

        //Parameterized Constructor
        Student s2 = new Student(55, 79, 95, 68);
        s2.display();

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

        //using accept function
        System.out.println("Enter: RollNo Marks1 Marks2
Marks3");
        s1.acceptInfo(sc.nextInt(), sc.nextInt(), sc.nextInt(),
sc.nextInt());
        s1.display();

        sc.close();

    }

}
```

Output :-

```
---Default constructor---
Roll No: 0 Percentage: 0.0 Grade: F
-----
---Parameterized constructor---
Roll No: 55 Percentage: 80.0 Grade: A+
-----
Enter: RollNo Marks1 Marks2 Marks3
21 65 71 83
---Inside AcceptInfo---
Roll No: 21 Percentage: 73.0 Grade: A
```

2. 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.

In pkgCode:-

```
package pkgCode;
public class Person {

    private String name, city;
    private int age;

    public Person() {
        System.out.println("---Inside Default cnstr---");
        name = "admin";
        age = 18;
        city = "Mumbai";
    }

    public Person(String name, int age, String city) {
        System.out.println("---Inside Parameterized cnstr---");
        this.name = name;
        this.age = age;
        this.city = city;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAge(int age) {
        this.age = age;
    }

    public void setCity(String city) {
        this.city = city;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getCity() {
        return city;
    }
}
```

```

    }
    public void display() {
        System.out.println("Name: " + name + "\nAge: " + age +
"\nCity: " + city);
    }
}

```

In pkgTester:-

```
package pkgTester;
```

```
import java.util.Scanner;
```

```
import pkgCode.Person;
```

```
public class PersonTester {
```

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

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

```
        Person p = new Person();
        p.display();
```

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

```
        p = new Person("Pranay", 24, "Thane");
        p.display();
```

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

```
        System.out.print("Enter New Name: ");
        p.setName(sc.next());
        System.out.println("Updated Name: " + p.getName());
```

```
        System.out.println("-----");
        System.out.print("Enter New Age: ");
        p.setAge(sc.nextInt());
        System.out.println("Updated Age: " + p.getAge());
```

```
        System.out.println("-----");
        System.out.print("Enter New City: ");
        p.setCity(sc.next());
        System.out.println("Updated City: " + p.getCity());
        System.out.println("-----");
        p.display();
```

```
        sc.close();
```

```
    }
```

```
}
```

Output :-

```
---Inside Default cnstr---
Name: admin
Age: 18
City: Mumbai
-----
---Inside Parameterized cnstr---
Name: Pranay
Age: 24
City: Thane
-----
Enter New Name: Mohan
Updated Name: Mohan
-----
Enter New Age: 25
Updated Age: 25
-----
Enter New City: Pune
Updated City: Pune
-----
Name: Mohan
Age: 25
City: Pune
```

3. 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.

In pkgCode:-

```
package pkgCode;

public class Date {

    private int day, month, year;

    public Date() {

        System.out.println("----- Inside Default Constructor -----");
        day = 1;
        month = 1;
        year = 0001;
    }

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

        System.out.println("----- Inside Parameterized Constructor -----");

        this.day = day;
        this.month = month;
        this.year = year;
    }

    public void display()
    {
        System.out.println("Date is: " + day + "/" + month + "/" + year);
    }
}
```

In pkgTester:-

```
package pkgTester;

import java.util.Scanner;

import pkgCode.Date;

public class DateTester {

    public static void main(String[] args) {

        // For default constructor
        Date d = new Date();
        d.display();

        // For parameterized constructor
        Scanner sc = new Scanner(System.in);
        System.out.println("Enetr Day Month Year: ");

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

        sc.close();
    }
}
```

Output :-

```
----- Inside Default Constructor -----
Date is: 1/1/1
Enetr Day Month Year:
21 03 2023
----- Inside Parameterized Constructor -----
Date is: 21/3/2023
```

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.

In pkgCode:-

```
package pkgCode;
public class Book {

    private String bookName, author;
    private int bookId;
    private Double price;

    public Book() {
        System.out.println("---Default Constructor---");

        this.bookName = "Ddata";
        this.author = "R. D. Sharma";
        this.bookId = 0007;
        this.price = 550.0;
    }

    public Book(String bookName, String author, int bookId, Double
price) {

        System.out.println("---Parameterized Constructor---");
        this.bookName = bookName;
        this.author = author;
        this.bookId = bookId;
        this.price = price;
    }

    public void setBookName(String bookName) {
        this.bookName = bookName;
    }

    public String getBookName() {
        return bookName;
    }

    public void setAuthor(String author) {
        this.author = author;
    }

    public String getAuthor() {
        return author;
    }
}
```

```
public void setBookId(int bookId) {
    this.bookId = bookId;
}

public int getBookId() {
    return bookId;
}

public Double getPrice() {
    return price;
}

public void setPrice(Double price) {
    this.price = price;
}

public void display() {
    System.out.println("BookName: "+bookName+"\nAuthor: "+
author+"\nBookId: "+bookId+"\nPrice: "+price);
}
}
```


In pkgTester:-

```
package pkgTester;

import java.util.Scanner;

import pkgCode.Book;

public class BookTester {

    public static void main(String[] args) {

        Book obj = new Book();
        obj.display();
        System.out.println("-----");

        System.out.println("Enter BookName Author BookId Price:");
        Scanner sc = new Scanner(System.in);

        obj = new Book(sc.next(), sc.next(), sc.nextInt(), sc.nextDouble());
        obj.display();
        System.out.println("-----");

        System.out.print("Enter Book Name to Update: ");
        obj.setBookName(sc.next());
        System.out.println("Updated Book Name : " + obj.getBookName());
        System.out.println("-----");

        System.out.print("Enter Author Name to Update: ");
        obj.setAuthor(sc.next());
        System.out.print("Updated Author Name : " + obj.getAuthor());
        System.out.println("\n-----");

        System.out.print("Enter Book Id to Update: ");
        obj.setBookId(sc.nextInt());
        System.out.print("Updated BookID : " + obj.getBookId());
        System.out.println("\n-----");

        System.out.print("Enter Book Price to Update: ");
        obj.setPrice(sc.nextDouble());
        System.out.print("Updated Book Price : " + obj.getPrice());
        System.out.println("\n-----");
        obj.display();

        sc.close();

    }

}
```

Output :-

```
---Default Constructor---  
BookName: Ddata  
Author: R. D. Sharma  
BookId: 7  
Price: 550.0  
-----  
Enter BookName Author BookId Price:  
Maths Shree 101 450  
---Parameterized Constructor---  
BookName: Maths  
Author: Shree  
BookId: 101  
Price: 450.0  
-----
```

```
-----  
Enter Book Name to Update: Physics  
Updated Book Name : Physics  
-----  
Enter Author Name to Update: Kalpesh  
Updated Author Name : Kalpesh  
-----  
Enter Book Id to Update: 102  
Updated BookID : 102  
-----  
Enter Book Price to Update: 650  
Updated Book Price : 650.0  
-----  
BookName: Physics  
Author: Kalpesh  
BookId: 102  
Price: 650.0
```

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.

In pkgCode:-

```
package pkgCode;

public class Point {

    private int x, y;

    public Point() {
        System.out.println("---Default Constructor---");
        x = 5;
        y = 5;
    }

    public Point(int x, int y) {
        System.out.println("---Parameterized Constructor---");
        this.x = x;
        this.y = y;
    }

    public void setX(int x) {
        this.x = x;
    }

    public int getX() {
        return x;
    }

    public void setY(int y) {
        this.y = y;
    }

    public int getY() {
        return y;
    }

    public void Display() {
        System.out.println("Value of x: " + x + " and Value of y:
" + y);
    }

}
```

In pkgTester:-

```
package pkgTester;

import java.util.Scanner;

import pkgCode.Point;

public class PointTester {

    public static void main(String[] args) {

        Point obj1 = new Point();
        obj1.Display();
        System.out.println("-----");
        // Parameterized constructor

        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the values of x and y: ");
        obj1 = new Point(sc.nextInt(), sc.nextInt());
        obj1.Display();
        System.out.println("-----");

        // Update value of x & y

        System.out.println("Enter value of x & y to update:");
        obj1.setX(sc.nextInt());
        obj1.setY(sc.nextInt());
        System.out.println("-----");

        // getter
        System.out.println("Updated value of x: " + obj1.getX() + " and y: " +
obj1.getY());

        sc.close();

    }

}
```

Output :-

```
---Default Constructor---  
Value of x: 5 and Value of y: 5  
-----  
Enter the values of x and y:  
15 25  
---Parameterized Constructor---  
Value of x: 15 and Value of y: 25  
-----  
Enter value of x & y to update:  
34 45  
-----  
Updated value of x: 34 and y: 45
```

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.

In pkgCode:-

```
package pkgCode;

public class ComplexNumber {

    private int real, img;

    public ComplexNumber() {
        System.out.println("---Default Cnstr---");
        real = 0;
        img = 0;
    }

    public ComplexNumber(int real, int img) {
        System.out.println("---Parameterized Cnstr---");
        this.real = real;
        this.img = img;
    }

    public int getReal() {
        return real;
    }

    public void setReal(int real) {
        this.real = real;
    }

    public int getImg() {
        return img;
    }

    public void setImg(int img) {
        this.img = img;
    }

    public void display() {
        System.out.println("Complex Number: " + real + "+i" +
img);
    }
}
```

In pkgTester:-

```
package pkgTester;

import java.util.Scanner;

import pkgCode.ComplexNumber;

public class ComplexNumberTester {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        ComplexNumber obj = new ComplexNumber();
        obj.display();
        System.out.println("-----");

        System.out.println("Enter real Number and imaginary numbers: ");
        obj = new ComplexNumber(sc.nextInt(), sc.nextInt());
        obj.display();
        System.out.println("-----");

        System.out.print("Enter real Number to update: ");
        obj.setReal(sc.nextInt());
        System.out.println("Updated real number: " + obj.getReal());
        System.out.println("-----");

        System.out.print("Enter imaginary Number to update: ");
        obj.setImg(sc.nextInt());
        System.out.println("Updated imaginary number: " + obj.getImg());
        System.out.println("-----");
        obj.display();

        sc.close();

    }

}
```

Output :-

```
---Default Cnstr---
Complex Number: 0+i0
-----
Enter real Number and imaginary numbers:
2 3
---Parameterized Cnstr---
Complex Number: 2+i3
-----
Enter real Number to update: 5
Updated real number: 5
-----
Enter imaginary Number to update: 7
Updated imaginary number: 7
-----
Complex Number: 5+i7
```


7. Create BankAccount application for operations like withdraw ,deposit and moneyTransfer.

//Create menu drive program for bank operations..

```
package com.test;
```

```
import java.util.Scanner;
```

```
import com.code.Account;
```

```
public class TesterAccount {
```

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

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

```
        System.out.println("Enter the no of accounts: ");
```

```
        int size= sc.nextInt();
```

```
        Account arr[]= new Account[size];
```

```
        Account act = null;
```

```
        int index= 0;
```

```
        int choice;
```

```
        System.out.println(" 1)Create an account 2)Deposit 3)Withdraw
```

```
4)Money Transfer 5)Check Balance 6)Exit...");
```

```
        do {
```

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

```
            choice= sc.nextInt();
```

```
            switch(choice) {
```

```
            case 1:// create account
```

```
                System.out.println("Enter Id: Name: Salary: ");
```

```
                act= new Account(sc.nextInt(), sc.next(), sc.nextDouble() );
```

```
                arr[index]= act;
```

```
                System.out.println("Account created successfully!!!");
```

```
arr[index].getDetails();
```

```
index++;
```

```
break;
```

case 2:

```
System.out.println("Enter Account Id: ");
```

```
int id=sc.nextInt();
```

```
double de;
```

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

```
{
```

```
if(arr[i].getActId()==id)
```

```
{
```

```
System.out.println("Account found!! Enter amount to Deposit");
```

```
de=sc.nextDouble();
```

```
arr[i].depositMoney(de);
```

```
break;
```

```
}//if
```

```
}//for
```

```
break;
```

case 3:

```
System.out.println("Enter Account Id: ");
```

```
id=sc.nextInt();
```

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

```
{
```

```
if(arr[i].getActId()==id)
```

```
{
```

```
System.out.println("Account found!! Enter amount to  
Withdraw");  
de=sc.nextDouble();  
arr[i].WithdrawMoney(de);
```

```

        break;

    }

}

break;

case 4:

// check sender exist or not


System.out.println("Enter senders ID: ");

int sendId=sc.nextInt();

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

{

    if(arr[i].getActId()==sendId)

    { sendId= i;

    System.out.println("sender id at index " + sendId);

    break;

    }

}

// check receiver exist or not

System.out.println("Enter receivers ID: ");

int recId=sc.nextInt();

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

{

    if(arr[i].getActId()==recId)

    { recId= i;

    System.out.println("receiver id at index "+ recId);

    break;

    }

}

```

```
        }  
        // amount  
        System.out.println("Enter amount to transfer:");  
        double amt= sc.nextDouble();  
        // final money transfer  
        arr[sendId].moneyTransfer(arr[recId], amt );  
        break;  
    case 6:  
        System.out.println("Exiting the code");  
        break;  
    default:  
        System.out.println("Invalid Input!!");  
    }  
}while(choice!=6);  
System.out.println("Thank You !!!");  
sc.close();  
}  
}
```

Output :-

```
Enter the no of accounts:
3
1) Create an account
2) Deposit
3) Withdraw
4) Money Transfer
5) Check Balance
6) Exit...
Enter choice:
1
Enter Id Name Balance
101 pranay_patil 49500
Account created successfully!!!
ID: 101 Name: pranay_patil Balance: 49500.0
```

```
Enter choice:
1
Enter Id Name Balance
102 nayan_joshi 35871
Account created successfully!!!
ID: 102 Name: nayan_joshi Balance: 35871.0
Enter choice:
1
Enter Id Name Balance
103 shree_gavali 92552
Account created successfully!!!
ID: 103 Name: shree_gavali Balance: 92552.0
```

```
Enter choice:
2
Enter Account Id:
101
Account found!!
Enter amount to Deposit:
500
Updated Balance: 50000.0
Enter choice:
3
Enter Account Id:
102
Account found!!
Enter amount to Withdraw:
71
Updated Balance: 35800.0
```

3.

```
Enter choice:
4
Enter your id:
103
Enter Receivers id:
101
Enter Money to transfer:
52
Amount transferred from 103 to 101
Balance remaining: 92500.0
```

4.

```
Enter choice:
5
Enter your Account id:
103
Your account balance is: 92500.0
Enter choice:
6
Exiting the code
```

5.

8. Create Student class with rollno, name, address. Write business logic for auto increment of rollno (don't accept roll no from user), Write parameterised constructor for accepting name and address only. Write getter and setter and display function

- a. Test Student class by creating 5 different objects and display all details (check rollno created automatically)
- b. Create an array of 5 students and show only names

Code:-

```
package com.java;

public class Q7_Student {

    private int rollNo;
    private String name, address;
    private static int rollCount;

    static {
        rollCount = 1000;
    }

    public Q7_Student(String name, String address) {
        rollCount++;
        this.rollNo = rollCount;
        this.name = name;
        this.address = address;
    }

    public String toString() {
        return "Roll No: " + rollNo + " Name: " + name +
"Address: " + address;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setAdd(String add) {
        this.address = add;
    }
}
```

```
    public String getName() {  
        return name;  
    }  
  
    public String getAdd() {  
        return address;  
    }  
}
```

Tester:-

```
package com.testers;

import java.util.Scanner;

import com.java.Q7_Student;

public class Q7_TesterStudent {

    public static void main(String[] args) {

        Q7_Student obj = new Q7_Student("Pranay ", "Pune");
        System.out.println(obj);

        Q7_Student obj1 = new Q7_Student("Mohan ", "Pune");
        System.out.println(obj1);

        System.out.println("----Setter---");
        obj.setName("Nayan");
        obj.setAdd("Thane");
        obj1.setName("Shree");
        obj1.setAdd("Nigdi");

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

        System.out.println("Updated Name:" + obj.getName());
        System.out.println("Updated Name:" + obj1.getName());
        System.out.println("Updated Address:" + obj.getAdd());
        System.out.println("Updated Address:" + obj1.getAdd());

        System.out.println("---Updated Details---");

        System.out.println(obj);
        System.out.println(obj1);

        System.out.println("----- Q7.1-----");
        Q7_Student[] arr = new Q7_Student[5];

        Q7_Student std = null;
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter Details of 5 student: ");
        for (int i = 0; i < arr.length; i++) {

            System.out.println("Enter name & Address");
            std = new Q7_Student(sc.next(), sc.next());
            arr[i] = std;

        }
        System.out.println("Details of 5 student: ");
    }
}
```



```

        for (Q7_Student e : arr)
        {
            System.out.println(e);

        }
        System.out.println("-----");
        System.out.println("Names of 5 students: ");
        for (int i = 0; i < arr.length; i++) {

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

        }

        sc.close();

    }

}

```

Output :-

```

Roll No: 1001 Name: Pranay Address: Pune
Roll No: 1002 Name: Mohan Address: Pune
----Setter---
---Getter---
Updated Name:Nayan
Updated Name:Shree
Updated Address:Thane
Updated Address:Nigdi
---Updated Details---
Roll No: 1001 Name: NayanAddress: Thane
Roll No: 1002 Name: ShreeAddress: Nigdi

```

7.0

```

----- Q7.1-----
Enter Details of 5 student:
Enter name & Address
Yogesh Amaravati
Enter name & Address
Hitesh Mumbai
Enter name & Address
Mahepal Jalgaon
Enter name & Address
Manoj Nagar
Enter name & Address
Piyush Dadar
Details of 5 student:
Roll No: 1003 Name: YogeshAddress: Amaravati
Roll No: 1004 Name: HiteshAddress: Mumbai
Roll No: 1005 Name: MahepalAddress: Jalgaon
Roll No: 1006 Name: ManojAddress: Nagar
Roll No: 1007 Name: PiyushAddress: Dadar

```

7.1

```

-----
Names of 5 students:
Yogesh
Hitesh
Mahepal
Manoj
Piyush

```

7.2

ASSIGNMENT 3:

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

1. Create a parameterised constructor to accept x & y co-ordinates.

2. Add public String show() --to return point's x & y co-ordinates

3. Add isEqual method to Point2D class : boolean returning method : must return true if both points are having same x & y co-ordinates or false otherwise.

4. Add a method , calculateDistance , to calc distance between 2 points (Hint : use distance formula) 5. Create a Driver class(for UI) , in the "tester" package

"TestPoints" , with main(..). Ask user , how many points to plot? Create suitable array.

6. Accept x & y co-ordinates for all the points n store it suitably.

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

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.

In Code :-

```
package com.cdac.geometry;

public class Point2D {

    protected int x;
    protected int y;

    public Point2D(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public String show() {
        return "X coordinate:" + x + "\nY coordinate: " + y;
    }

    public boolean isEqual(Point2D a) {
        if (this.x == a.x && this.y == a.y) {
            return true;
        } else {
            return false;
        }
    }

    public double calculateDistance(Point2D a) {
        double distance = Math.sqrt((this.x - a.x) * (this.x - a.x) + (this.y - a.y) * (this.y - a.y));
        return distance;
    }
}
```

```
}  
  
}
```

In Tester :-

```
package com.cdac.TestPoints;  
  
import java.util.Scanner;  
  
import com.cdac.geometry.Point2D;  
  
public class Driver {  
  
    public static void main(String[] args) {  
  
        Scanner in = new Scanner(System.in);  
  
        System.out.println("How many X & Y coordinates do you  
wish to plot: ");  
        int num = in.nextInt();  
  
        Point2D[] point = new Point2D[num];  
  
        for (int i = 0; i < num; i++) {  
            System.out.println("\nEnter the " + (i + 1) + " X  
and Y coordinates of your Points: ");  
            point[i] = new Point2D(in.nextInt(), in.nextInt());  
        }  
  
        for (Point2D p : point) {  
            if (p != null) {  
                System.out.println("\nX and Y co-ordinates  
are: \n" + p.show());  
            }  
        }  
  
        System.out.println("-----");  
        System.out.println("\nEnter 2 more Indices: ");  
        System.out.println("X and Y coordinates of 1st Index: ");  
        Point2D a = new Point2D(in.nextInt(), in.nextInt());  
  
        System.out.println("X and Y coordinates of 2nd Index: ");  
        Point2D b = new Point2D(in.nextInt(), in.nextInt());  
  
        boolean ans = a.isEqual(b);
```

```

        if (ans == true) {
            System.out.println("Your Indices are same");
        } else {
            System.out.println("Your Indices are different");
            double distance =
Math.round(a.calculateDistance(b));
            System.out.println("Distance between your
points is: " + distance + " Units");

            in.close();
        }
    }
}

```

Output :-

```

How many X & Y coordinates do you wish to plot:
2

Enter the 1 X and Y coordinates of your Points:
8 3

Enter the 2 X and Y coordinates of your Points:
4 9

X and Y co-ordinates are:
X coordinate:8
Y coordinate: 3

X and Y co-ordinates are:
X coordinate:4
Y coordinate: 9
-----

Enter 2 more Indices:
X and Y coordinates of 1st Index:
4 9
X and Y coordinates of 2nd Index:
4 9
Your Indices are same

```

For same Indices

```

How many X & Y coordinates do you wish to plot:
2

Enter the 1 X and Y coordinates of your Points:
8 3

Enter the 2 X and Y coordinates of your Points:
4 9

X and Y co-ordinates are:
X coordinate:8
Y coordinate: 3

X and Y co-ordinates are:
X coordinate:4
Y coordinate: 9
-----

Enter 2 more Indices:
X and Y coordinates of 1st Index:
8 5
X and Y coordinates of 2nd Index:
5 8
Your Indices are different
Distance between your points is: 4.0 Units

```

For different Indices

2. Arrange Fruit,Apple,Orange,Mango in inheritance hierarchy.

Use tight encapsulation. Properties (instance variables) : colour : String , weight : double , name:String, fresh : boolean.

Add suitable constructor. Override toString correctly to return state of all fruits (return only : name ,colour , weight) 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"

Add specific functionality , in the sub classes

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

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

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

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

Create java application FruitBasket , with main method , as a tester.

Prompt user for the basket size n create suitable data structure. 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, colour, 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))

9. Exit

In Fruit :-

```
package com.app.fruits;
```

```
public class Fruit {
```

```
    protected String colour;
```

```
    protected double weight;
```

```
    protected String name;
```

```

    protected boolean fresh;

    public Fruit(String name, double weight, String colour,
boolean fresh) {
        this.name = name;
        this.weight = weight;
        this.colour = colour;
        this.fresh = fresh;
    }

    @Override
    public String toString() {
        return "\nName: " + this.name + "\nColour: " + this.colour +
"\nWeight: " + this.weight;
    }

    public String taste() {
        System.out.println("\nCan you identify the taste?");
        return "No specific taste";
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setWeight(double weight) {
        this.weight = weight;
    }

    public double getWeight() {
        return weight;
    }

    public void setColour(String colour) {
        this.colour = colour;
    }

    public String getColour() {
        return colour;
    }

    public void setFresh(boolean fresh) {

```

```

        this.fresh = fresh;
    }

    public boolean getFresh() {
        return fresh;
    }
}

```

In Apple :-

```

package com.app.fruits;

public class Apple extends Fruit {

    public Apple(String name, double weight, String colour,
boolean fresh) {
        super(name, weight, colour, fresh);
        this.name = name;
        this.weight = weight;
        this.colour = colour;
        this.fresh = fresh;
    }

    @Override
    public String taste() {
        System.out.println("\nCan you identify the taste?");
        return "Sweet and Sour";
    }

    public void jam() {
        System.out.println("Name of Fruit: " + this.name + "\n---
Making Jam---");
    }
}

```

In Mango :-

```

package com.app.fruits;

public class Mango extends Fruit {
    public Mango(String name, double weight, String colour,
boolean fresh) {
        super(name, weight, colour, fresh);
        this.name = name;
        this.weight = weight;
    }
}

```

```

        this.colour = colour;
        this.fresh = fresh;
    }

    @Override
    public String taste() {
        System.out.println("\nCan you identify the taste?");
        return "Sweet";
    }

    public void pulp() {
        System.out.println("\nName of Fruit: " + name + "Colour: " + colour + "\n---Extracting Pulp---");
    }
}

```

In Orange :-

```

package com.app.fruits;

public class Orange extends Fruit {

    public Orange(String name, double weight, String colour, boolean fresh) {
        super(name, weight, colour, fresh);
        this.name = name;
        this.weight = weight;
        this.colour = colour;
        this.fresh = fresh;
    }

    @Override
    public String taste() {
        System.out.println("\nCan you identify the taste?");
        return "No specific taste";
    }

    public void juice() {
        System.out.println("\nFruit name: " + name + "Weight" + weight + "\n---Extracting juice---");
    }
}

```


In FruitBasket :-

```
package com.app.fruits;

import java.util.Scanner;

public class FruitBasket {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        System.out.println("Enter the number of Fruits in your
Basket: ");
        int num = in.nextInt();

        System.out.println("Select from the menu mentioned
below:");
        System.out.println("1. Add Mango");
        System.out.println("2. Add Orange");
        System.out.println("3. Add Apple");
        System.out.println("4. Display Names of all Fruits in the
Basket");
        System.out.println("5. Display Name, Colour, Weight,
Taste of all Fresh Fruits in the basket");
        System.out.println("6. Mark a Fruit as Stale");
        System.out.println("7. Mark all Sour Fruits as Stale");
        System.out.println("8. Make Mango Pulp, Orange Juice or
Apple Jam");

        Fruit[] fruit = new Fruit[num];
        int index = 0;
        int choice;

        do {
            System.out.println("\nEnter your choice");
            choice = in.nextInt();

            switch (choice)
            {

                case 1:
                    if (index < (num)) {
                        System.out.println("\nEnter the details
of your Fruit(Mango): ");
                        System.out.println("Name: \nWeight:
\nColour: \nFresh(Boolean): ");
```

```

        fruit[index] = new Mango(in.next(),
        in.nextDouble(), in.next(),
        in.nextBoolean());
        index++;
    } else {
        System.out.println("Your Basket is
full");
    }
    break;

```

```

case 2:
    if (index < (num)) {
        System.out.println("\nEnter the details
of your Fruit(Orange): ");
        System.out.println("Name: \nWeight:
\nColour: \nFresh(Boolean): ");
        fruit[index] = new Orange(in.next(),
        in.nextDouble(), in.next(),
        in.nextBoolean());
        index++;
    } else {
        System.out.println("Your Basket is
full");
    }
    break;

```

```

case 3:
    if (index < (num)) {
        System.out.println("\nEnter the details
of your Fruit(Apple): ");
        System.out.println("Name: \nWeight:
\nColour: \nFresh(Boolean): ");
        fruit[index] = new Apple(in.next(),
        in.nextDouble(), in.next(),
        in.nextBoolean());
        index++;
    } else {
        System.out.println("Your Basket is
full");
    }
    break;

```

```

case 4:
    System.out.println("\nBelow are the names of
all the fruits in your Basket: ");
    for (Fruit f : fruit) {

```

```

        if (f != null) {
            System.out.println("Name of Fruit:
                                " + f.getName());
        }
    }
    break;

case 5:
    System.out.println("\nBelow are the Details of
all the          Fresh Fruits in your Baskets:
");
    for (int i = 0; i < index; i++)
        if (fruit[i].getFresh()) {

System.out.println(fruit[i].toString());

System.out.println(fruit[i].taste());
        }
    break;

```

```

case 6:
    System.out.println("Select which Fruit you
want to mark as Stale:");
    for (int i = 0; i < index; i++) {
        System.out.println((i + 1) + ". " + "Name
of Fruit: " + fruit[i].getName());
    }

    System.out.println("Enter your selection: ");
    int sel = in.nextInt();
    fruit[sel - 1].setFresh(false);
    System.out.println(fruit[sel - 1].getName() +
" has been marked as Stale");
    break;

```

```

case 7:
    System.out.println("The following Fruits that
were Sour have been marked as Stale");
    for (int i = 0; i < index; i++) {
        if (fruit[i].taste().equals("Sour") ||
fruit[i].taste().equals("sour")) {
            fruit[i].setFresh(false);

```

```

        System.out.println(fruit[i].getName());
    }
}

break;

case 8:
    System.out.println("Select the fruit of your
choice to perform Fruit-specific function: ");

    for (int i = 0; i < index; i++) {
        System.out.println((i + 1) + ". " + "Name
of Fruit: " + fruit[i].getName());
    }

    System.out.println("Enter your selection: ");
    sel = in.nextInt();

    if (fruit[(sel - 1)] instanceof Apple) {
        ((Apple) fruit[(sel - 1)]).jam();
    }

    if (fruit[(sel - 1)] instanceof Mango) {
        ((Mango) fruit[(sel - 1)]).pulp();
    }

    if (fruit[(sel - 1)] instanceof Orange) {
        ((Orange) fruit[(sel - 1)]).juice();
    }

    break;

case 10:
    break;

default:
    System.out.println("Invalid Selection, Please
select from the menu below:");
    System.out.println("1. Add Mango");
    System.out.println("2. Add Orange");
    System.out.println("3. Add Apple");
    System.out.println("4. Display Names of all
Fruits in the Basket");
    System.out.println("5. Display Name, Colour,
Weight, Taste of all Fresh Fruits in the
basket");
    System.out.println("6. Mark a Fruit as
Stale");

```

```

        System.out.println("7. Mark all Sour Fruits as
Stale");

        System.out.println("8. Make Mango Pulp, Orange
Juice or Apple Jam");

        break;

    }
} while (choice != 10);

in.close();
}
}

```

Output :-

```

Enter the number of Fruits in your Basket:
4
Select from the menu mentioned below:
1. Add Mango
2. Add Orange
3. Add Apple
4. Display Names of all Fruits in the Basket
5. Display Name, Colour, Weight, Taste of all Fresh Fruits in the basket
6. Mark a Fruit as Stale
7. Mark all Sour Fruits as Stale
8. Make Mango Pulp, Orange Juice or Apple Jam

Enter your choice
1

Enter the details of your Fruit(Mango):
Name: Weight: Colour: Fresh(Boolean):
Mango 35 Yellow true

Enter your choice
2

Enter the details of your Fruit(Orange):
Name: Weight: Colour: Fresh(Boolean):
Orange 20 Orange true

Enter your choice
3

Enter the details of your Fruit(Apple):
Name: Weight: Colour: Fresh(Boolean):
Apple 30 Red true

Enter your choice
1

Enter the details of your Fruit(Mango):
Name: Weight: Colour: Fresh(Boolean):
Mango 30 Yellow false

Enter your choice
4

Below are the names of all the fruits in your Basket:
Name of Fruit: Mango
Name of Fruit: Orange
Name of Fruit: Apple
Name of Fruit: Mango

Enter your choice
8
Select the fruit of your choice to perform Fruit-specific function:
1. Name of Fruit: Mango
2. Name of Fruit: Orange
3. Name of Fruit: Apple
4. Name of Fruit: Mango
Enter your selection:
2

Fruit name: OrangeWeight20.0
---Extracting juice---

```

1.

```

Enter your choice
5

Below are the Details of all the Fresh Fruits in your Baskets:

Name: Mango
Colour: Yellow
Weight: 35.0

Can you identify the taste?
Sweet

Name: Orange
Colour: Orange
Weight: 20.0

Can you identify the taste?
No specific taste

Name: Apple
Colour: Red
Weight: 30.0

Can you identify the taste?
Sweet and Sour

```

```

Enter your choice
6
Select which Fruit you want to mark as Stale:
1. Name of Fruit: Mango
2. Name of Fruit: Orange
3. Name of Fruit: Apple
4. Name of Fruit: Mango
Enter your selection:
4
Mango has been marked as Stale

```

2.

3. Solve this business Scenario. Fresh business scenario to apply inheritance , polymorphism n abstraction to Emp based organisation scenario.

Create Emp based organization structure — Emp , Mgr , Worker. All of above classes must be in --com.app.org

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

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 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. getHourlyRate of the worker -- add a new method to return hourly rate of a worker. (getter). Organise classes in inheritance hierarchy.

NOTE : toString method SHOULD NOT include the net salary of the employee Write TestOrganisation 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 /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 centralised manner (This can be added later!)

10. Exit

Emp :-

```
package com.app.org;
```

```
public class Emp {  
    protected int empId;  
    protected String firstName;  
    protected String lastName;  
    protected int deptId;  
    protected double basic;  
  
    private static int counter;
```

```

        static {
            counter = 2303000;
        }

        public Emp(String firstName, String lastName, int deptId, double
        basic) {
            counter++;
            this.empId = counter;
            this.firstName = firstName;
            this.lastName = lastName;
            this.deptId = deptId;
            this.basic = basic;
        }

        @Override
        public String toString() {
            return "\nEmployee name: " + firstName + " " + lastName +
            "\nEmployee Id: " + empId + "\nDepartment Id: "
            + deptId;
        }

        public void computeNetSalary() {
            System.out.println("Salary: ₹" + basic);
        }

        public void setBasic(double basic) {
            this.basic = basic;
        }

        public double getBasic() {
            return basic;
        }
    }
}

```

Manager :-

```

package com.app.org;

public class Mgr extends Emp {

    private double perfBonus;

    public Mgr(String firstName, String lastName, int deptId,
    double basic, double perfBonus) {
        super(firstName, lastName, deptId, basic);
    }
}

```

```

        this.perfBonus = perfBonus;
    }

    @Override
    public String toString() {
        return "\nEmployee name: " + firstName + " " + lastName +
            "\nEmployee Id: " + empId + "\nDepartment Id: "
                + deptId;
    }

    @Override
    public void computeNetSalary() {
        System.out.println("Salary: ₹" + basic + perfBonus);
    }

    public void getPerfBonus() {
        System.out.println("Performance Bonus: ₹" + perfBonus);
    }
}

```

Worker :-

```

package com.app.org;

public class Worker extends Emp {

    private int hoursWorked;
    private int hourlyRate;

    public Worker(String firstName, String lastName, int deptId,
        double basic, int hoursWorked, int hourlyRate) {
        super(firstName, lastName, deptId, basic);
        this.hourlyRate = hourlyRate;
        this.hoursWorked = hoursWorked;
    }

    @Override
    public String toString() {
        return "\nEmployee name: " + firstName + " " + lastName +
            "\nEmployee Id: " + empId + "\nDepartment Id: "
                + deptId;
    }

    @Override
    public void computeNetSalary() {

```



```

        String f_name = in.nextLine();
        in.nextLine();
        System.out.println("Last Name:");
        String l_name = in.nextLine();
        System.out.println("Department Id:");
        int deptId = in.nextInt();
        System.out.println("Basic Salary:");
        double sal = in.nextDouble();
        System.out.println("Performance Bonus:");
        double perfBonus = in.nextDouble();
        employee[index] = new Mgr(f_name, l_name,
        deptId, sal, perfBonus);
        System.out.println("Manager Added

Successfully..!");
        index++;
    } else {
        System.out.println("Your organization is

full");
    }
    break;

case 2:
    if (index < size) {
        System.out.println("\nEnter the details
of your Worker: ");
        System.out.println("First Name:");
        String f_name = in.nextLine();
        in.nextLine();
        System.out.println("Last Name:");
        String l_name = in.nextLine();
        System.out.println("Department Id:");
        int deptId = in.nextInt();
        System.out.println("Basic Salary:");
        double sal = in.nextDouble();
        System.out.println("Hours Worked:");
        int hours_worked = in.nextInt();
        System.out.println("Hourly Rate:");
        int hourly_rate = in.nextInt();
        employee[index] = new Worker(f_name,
        l_name, deptId, sal, hours_worked,
        hourly_rate);
        System.out.println("Worker Added

Successfully..!");
        index++;
    } else {
        System.out.println("Your organization is

full");
    }
    break;

```

```

        case 3:
            System.out.println("\nEmployees of your
Organization:");
            for (int i = 0; i < index; i++) {

                System.out.println(employee[i].toString());
            }
            break;

        case 4:
            System.out.println("\nSelect who's Base Salary
has to be changed: ");
            for (int i = 0; i < index; i++) {
                System.out.print((i + 1) + ".");

                System.out.println(employee[i].toString());
            }
            System.out.print("\nEnter your choice: ");
            int ch = in.nextInt();

            if (ch > (index + 1)) {
                System.out.println("\nInvalid
selection");

                break;
            } else {
                System.out.println("Enter new Basic
Salary: ");

                ch--;
                employee[ch].setBasic(in.nextDouble());

                System.out.println("New salary is ₹" +
employee[ch].getBasic());
                break;
            }
        case 10:
            System.out.println("Exiting...");
            break;

        default:
            System.out.println("\nInvalid Selection:
Please select from the options mentioned
below:");
            System.out.println("1. Hire Manager");
            System.out.println("Hire Worker");
            System.out.println("Display information of all
the Employees in the Organization");
            break;
    }
}

```

```

        } while (choice != 10);

        in.close();

    }
}

```

Output :-

Enter the number of Employees in your Organization
2

Select your option of preference:
1. Hire Manager
2. Hire Worker
3. Display information of all the Employees in the Organization
4. Update Basic Salary
10. Exit

Enter your choice:
1

Enter the details of your Manager:
First Name:
Pranay
Last Name:
Patil
Department Id:
101
Basic Salary:
45000
Performance Bonus:
15000
Manager Added Successfully..!

Enter your choice:
2

Enter the details of your Worker:
First Name:
Mohan
Last Name:
Talvar
Department Id:
102
Basic Salary:
20000
Hours Worked:
8
Hourly Rate:
300
Worker Added Successfully..!

Enter your choice:
3

Employees of your Organization:

Employee name: Patil
Employee Id: 2303001
Department Id: 101

Employee name: Talvar
Employee Id: 2303002
Department Id: 102

Enter your choice:
4

Select who's Base Salary has to be changed:

1.
Employee name: Patil
Employee Id: 2303001
Department Id: 101
2.
Employee name: Talvar
Employee Id: 2303002
Department Id: 102

Enter your choice: 2
Enter new Basic Salary:
15000
New salary is ₹15000.0

Enter your choice:
10
Exiting...

Assignment no 04

1. Solve the following:

1. Create abstract class Shape --state : x, y Abstract Method --public double area(); public String toString() : to return 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

Shape :-

```
public abstract class Shape {
    protected String x;
    protected String y;

    public abstract double area();

    public String toString() {
        return x + y;
    }
}
```

Circle :-

```
public class Circle extends Shape {
    private int radius;

    public double area() {
        double area = 3.14 * radius * radius;
        return area;
    }

    public void setRadius(int radius) {
        this.radius = radius;
    }
}
```

Rectangle :-

```
public class Rectangle extends Shape {  
    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 :-

```
public class Square extends Shape {  
    private int side;  
  
    public double area() {  
        double area = side * side;  
        return area;  
    }  
  
    public void setSide(int side) {  
        this.side = side;  
    }  
}
```

ShapeFactory :-

```
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;
    }

}

```

Tester :-

```

import java.util.Scanner;

public class Tester {
    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("Loop Endss...!!");
in.close();
}
}

```

Output :-

Randomly selected shapes by autogenerated random numbers in switch case and in Loop get terminate when that number is equal to 5

```

Shape generated: Circle
Enter the radius of the Circle:
2
Area of your circle is: 12.56

```

```

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: Square
Enter the length of the side of the Square:
4
Area of the Square is: 16.0

```

```

Shape generate: Rectangle
Enter the Width of the Rectangle:
2
Enter the Height of the Rectangle:
6
Area of your Rectangle is: 12.0

```

```

Shape not formed. Please try again!

Shape not formed. Please try again!
Loop 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.

IPrintable :-

```
public interface IPrintable {  
  
    void print();  
  
}
```

Employee :-

```
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 :-

```
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 :-

```
public abstract class Shape implements IPrintable {  
  
    public abstract double area();  
  
    @Override
```

```

        public void print() {

        }

    }

```

Rectangle :-

```

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 :-

```

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 :-

```

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 mohan 26000
Employee Details
ID:101 Name:mohan Salary:26000.0

Enter Date in dd/mm/yyyy format
23 02 2023
Date: 23/2/2023
```

1.

```
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:
3
5
Area Of Rectangle: 15.0
Enter choice:
2
Enter the value of radius to calculate area of circle:
2
Area of Circle:12.56
Enter choice:
3
Exiting the code....
```

2.