Topic : Programming Fundamentals		
I-Practice (Mandatory)		
Question	Solution	Remarks
FACTORIAL OF GIVEN NUMBER	**Factorial of a Number:** **Read Number; Set Fact=1, i=1** **WHILE i<=number** **Fact=Fact*i** **i=i+1** **END-WHILE** **WRITE Fact**	
FIBONACCI SEQUENCE UPTO TO THE GIVEN NUMBER	**Fibonacci Series:** **Read number,Set variables j=0,k=1, l=1** **FOR: (integer count is less than number, INCREMENT count every loop) DO** **PRINT integer j** **SET j=k, k=l, l=j+k** **END FOR:**	
REVERSE A NUMBER	**It has a Diagram**	
SWAPING TWO NUMBERS	**It has a Diagram**	
ELIGIBLE OR NOT	**It has a Diagram**	
SUM OF ELEMENTS IN ARRAY	**It has a Diagram**	
I-Practice (Bridges)		
Question	Solution	Remarks

GREATEST COMMON DIVISOR	**GCD of two numbers:**
	Declare n1 , n2 , gcd = 1, i
	Read n1 and n2
	FOR i = 1; i <= n1 && i <= n2; ++i THEN
	IF n1 % i == 0 && n2 % i == 0 THEN
	gcd = i
	END IF
	END FOR
	WRITE gcd	

MAXIMUM AMONG THREE NUMBERS	**Maximum Among three numbers:** **Read three numbers n1,n2,n3** **IF: n1 > n2 AND n1 > n3 THEN** **Write n1 as greater** **ELSE-IF: n2 > n3 THEN** **Write n2 as greater** **ELSE:** **Write n3 is greater** **END-IF**	
EVEN OR ODD	**It has a Diagram**	
I-Practice (Hots)		
Question	Solution	Remarks

PALIDROME	**Palindrome(number):**
	SET reverse = 0 and temp=number
	WHILE (number > 0):
	SET lastDigit = number % 10
	SET reverse = (reverse * 10) + lastDigit and SET number = number / 10
	ENDWHILE
	IF (temp == reverse) THEN
	PRINT 'PALINDROME'
	ELSE
	PRINT 'NOT A PALINDROME'
	ENDIF
	END	

MAXIMUM ELEMENT IN AN ARRAY	**ArrayMaxElement(arr, N):**
	SET i=1 and max=arr[0]
	WHILE (i<N):
	IF (arr[i]>max) THEN
	SET max=arr[i]
	ENDIF
	SET i=i+1
	ENDWHILE
	PRINT max
	END	

THE HAUNTED HOUSE	**It has a Diagram**	NOT ACCEPTED THE GIVEN DIAGRAM
Topic : Opp Using Java		
I-Practice (Mandatory)		
Question	Solution	Remarks
	Quiz	
I-Practice (Bridges)		
Question	Solution	Remarks

	Quiz	
I-Practice (Hots)		
Question	Solution	
	Quiz	
Topic : Basic Elements of Java		
I-Practice (Mandatory)		
Question	Solution	Remarks

Customized Welcome Message	import java.util.*;
	
	class Main {
	  public static void main(String[] args) {
	
	  Scanner sc = new Scanner(System.in);
	  System.out.println("Enter your name");
	  String name = sc.nextLine();
	  sc.close();
	
	  System.out.println("Hello "+name+" ! Welcome to Amphi Event Management System");
	
	  }
	}	

Total Expenses for the Event	import java.util.*;
	import java.io.*;
	import java.text.*;
	
	class Main{
	  public static void main(String[] args) throws Exception{
	
	  Scanner sc = new Scanner(System.in);
	  System.out.println("Enter branding expenses");
	  int Braex = sc.nextInt();
	  System.out.println("Enter travel expenses");
	  int traex = sc.nextInt();
	  System.out.println("Enter food expenses");
	  int foex = sc.nextInt();
	  System.out.println("Enter logistics expenses");
	  int logex = sc.nextInt();
	  sc.close();
	
	  DecimalFormat nf = new DecimalFormat("0.00");
	
	  float total = Braex+traex+foex+logex;
	  float brandper = (Braex/total)*100;
	  float travelper = (traex/total)*100;
	  float foodper = (foex/total)*100;
	  float logper = (logex/total)*100;
	
	  System.out.println("Total expenses : Rs."+nf.format(total));
	  System.out.println("Branding expenses percentage : "+nf.format(brandper)+"%");
	  System.out.println("Travel expenses percentage : "+nf.format(travelper)+"%");
	  System.out.println("Food expenses percentage : "+nf.format(foodper)+"%");
	  System.out.println("Logistics expenses percentage : "+nf.format(logper)+"%");
	
	  }
	}	

Thrill ride	import java.util.*;
	import java.io.*;
	
	class Main{
	  public static void main(String[] args) throws Exception{
	
	  Scanner sc = new Scanner(System.in);
	  int age = sc.nextInt();
	  sc.close();
	
	  if(age>0){
	  if(age < 15){
	
	  System.out.println("Not Allowed");
	
	  }
	  else if(age >60){
	
	  System.out.println("Not Allowed");
	
	  }
	  else{
	
	  System.out.println("Allowed");
	  }
	  }
	  }
	}	

CHARACTER PATTERN 3	import java.util.*;
	import java.io.*;
	
	class Main{
	  public static void main(String[] args) throws Exception{
	
	  Scanner sc = new Scanner(System.in);
	  int n = sc.nextInt();
	  sc.close();
	
	  for(int i=0;i<=n;i++){
	
	  for(int j=0;j<i;j++){
	  System.out.print("*");
	  }	

	System.out.println(); } } }	

S3P16-SERIES1	import java.util.Scanner;
	public class Main{
	public static void main(String[] args){
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	int status = 1;
	int num = 3;
	
	if(n>=1) {
	System.out.print(2+" ");
	}
	
	for(int i=2;i<=n;) {
	
	for(int j=2;j<=Math.sqrt(num);j++) {
	
	if(num%j==0) {
	status = 0;
	break;
	}
	
	}
	
	if(status !=0) {
	System.out.print(num+" ");
	i++;
	}
	
	status = 1;
	num++;
	}
	}
	}	

I-Practice (Bridges)		
Question	Solution	Remarks

S1P1-WELCOME MESSAGE	public class Main{
	public static void main(String[] args){
	//Fill your code
	System.out.println("Welcome to Amphi Event Management System");
	}
	}	

S2P2-TICKET TYPE	import java.util.Scanner;
	public class Main{
	public static void main(String[] args){
	
	Scanner sc = new Scanner(System.in);
	int age = sc.nextInt();
	
	if(age <15){
	System.out.println("Child Ticket");
	}
	else{
	System.out.println("Adult Ticket");
	}
	}
	}	

S3P3-LUCKY PAIRS	import java.util.*;
	public class Main{
	public static void main(String[] args){
	
	Scanner sc = new Scanner(System.in);
	int A = sc.nextInt();
	int B = sc.nextInt();
	int N = sc.nextInt();
	sc.close();
	
	int even =B;
	int odd = A;
	
	for(int i=1;i<=N;i++){
	
	if(i%2 ==1){
	
	odd = odd*2;
	
	}
	else{
	
	even = even*2;
	
	}
	}
	
	System.out.println(odd+even);
	
	}
	}	

CHARACTER PATTERN 1A	import java.util.Scanner;
	public class Main {
	
	public static void main(String[] args) {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();	

	for(int i=0;i<n;i++){
	
	for(int j=0;j<n;j++){
	
	System.out.print("*");
	
	}
	
	System.out.println();
	}
	}
	
	}	

CHARACTER PATTERN 1B	import java.util.Scanner;
	public class Main {
	
	public static void main(String[] args) {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	int i =0;
	while(i<n){
	int j=0;
	while(j<n){
	System.out.print("*");
	j++;
	}
	
	System.out.println();
	i++;
	
	}
	}
	}
	
	}	

ALPHABET PATTERNS 1	import java.util.Scanner;
	public class Main {
	
	public static void main(String[] args) {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	int value = 65;
	
	for(int i=0;i<n;i++){
	
	for(int j=0;j<=i;j++)
	{
	System.out.print((char)(value+j)+"");
	}
	System.out.println();
	}
	
	}
	
	}	

ALPHABET PATTERNS 4	import java.util.Scanner;
	public class Main {
	
	public static void main(String[] args) {
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	int value = 65;
	
	for(int i=0;i<n;i++){
	
	for(int j=0;j<=i;j++)
	{
	System.out.print((char)(value+n-1 -j)+"");
	}
	System.out.println();
	}
	}
	
	}	

DAY 11 CRITERIA	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	int count = 4;
	int m = 6;
	int x = 6;
	
	int arr[] = new int[n];
	
	for(int i =0;i<n;i++){
	
	if(i==0){
	arr[i] = 24;
	}
	else if(i==1){
	arr[i] = 60; }
	else{
	arr[i] =arr[i-1]+(m*(x+count));
	x= x+count;
	count = count +1;	

	} System.out.print(arr[i]+" "); } } }	

DAY 13 CRITERIA	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	int count = 3;
	int m = 4;
	int arr[] = new int[n];
	
	for(int i =0;i<n;i++){
	
	if(i==0){
	arr[i] = 5;
	}
	else{
	
	arr[i] =arr[i-1]+(m*count);
	count = count +2;
	}
	System.out.print(arr[i]+" ");
	}
	}
	}	

I-Practice (Hots)		
Question	Solution	Remarks

S1P13-PRANAV AND CHANGE	import java.util.Scanner;
	class Main {
	public static void main(String[] args) {
	
	Scanner sc =new Scanner(System.in);
	int amount = sc.nextInt();
	sc.close();
	
	int Hundrend = amount/100;
	int Fifty = (amount%100)/50;
	int Ten =((amount%100)%50)/10;
	int Five =(((amount%100)%50)%10)/5;
	int Two = ((((amount%100)%50)%10)%5)/2 ;
	int One = ((((amount%100)%50)%10)%5)%2;
	
	System.out.println(Hundrend+Fifty+Ten+Five+Two+One);
	
	}
	}	

S3P17-SERIES2	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	int first = 20;
	int m = 4;
	int x = 10;
	int count = 0;
	int arr[] = new int[n];
	
	for(int i =0;i<n;i++){
	
	if(i==0){
	arr[i] = first;
	}
	else{
	arr[i] =arr[i-1]+(m*(x+count));
	count = count +1;
	}
	System.out.print(arr[i]+" ");
	}
	}
	}	

AF_PATTERN5	import java.util.Scanner;
	public class Main{
	public static void main(String[] args){
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	for(int i=1;i<=n;i++){
	
	for(int j=1;j<=(n+1);j++){
	
	if((i%2==1)&&(j==(n+1))){
	
	System.out.print((i+(n-2))+"");
	}
	else if((i%2==0)&&(j==1)){
	
	System.out.print((i+(n-2))+" ");
	}
	else if(j==(n+1)){	

	System.out.print(i);
	}
	else {
	
	System.out.print(i +" ");
	}
	}
	System.out.println();
	}
	
	}
	}	

PATTERN 1	import java.util.Scanner;
	public class Main{
	public static void main(String[] args){
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	for(int i=1;i<=n;i++){
	
	for(int j=1;j<=n;j++){
	
	if((i==1)||(i==n)||(i==((n/2)+1))){
	if(j==n){
	System.out.print(0);
	}
	else{
	System.out.print(1);
	}
	}
	else{
	if(j==1){
	System.out.print(1);
	}
	else if(j==n){
	System.out.print(1);
	}
	else{
	System.out.print(0);
	}
	}
	}
	System.out.println();
	}
	
	}
	}	

PATTERN 2	import java.util.Scanner;
	public class Main{
	public static void main(String[] args){
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	for(int i=1;i<=n;i++){
	
	for(int j=1;j<=n;j++){
	
	if((i==1)||(i==n)){
	if((j==n)||(j==1)){
	System.out.print(0);
	}
	else{
	System.out.print(1);
	}
	}
	else if((i==2)||(i==n-1)){
	if((j==n)||(j==1)){
	System.out.print(1);
	}
	else{
	System.out.print(0);
	}
	}
	else{
	if(j==1){
	System.out.print(1);
	}
	else{
	System.out.print(0);
	}
	}
	}
	System.out.println();
	}
	}
	}	

PATTERN 3	import java.util.Scanner;
	public class Main{
	public static void main(String[] args){
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	for(int i=1;i<=n;i++){
	
	for(int j=1;j<=n;j++){
	
	if(i==1){	

	System.out.print(1);
	}
	else if(i==(n-1)){
	if((j==1)||(j==((n/2)+1))){
	System.out.print(1);
	}
	else{
	System.out.print(0);
	}
	}
	else if(i==n){
	if((j==1)||(j>(n/2))){
	System.out.print(0);
	}
	else{
	System.out.print(1);
	}
	}
	else{
	if(j==((n/2)+1)){
	System.out.print(1);
	}
	else{
	System.out.print(0);
	}
	}
	}
	System.out.println();
	}
	}
	}	

DAY 1 CRITERIA	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	int m = 8;
	int count = 1;
	int arr[] = new int[n];
	
	for(int i =0;i<n;i++){
	
	if(i==0){
	arr[i] = 1;
	}
	else{
	
	arr[i] =arr[i-1]+(m*count);
	
	if(i%2==0){
	count = count +1;
	
	}else{
	count = count;
	}
	}
	System.out.print(arr[i]+" ");
	}
	}
	}	

DAY 2 CRITERIA	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	int count = 3;
	int arr[] = new int[n];
	
	for(int i =0;i<n;i++){
	
	if(i==0){
	arr[i] = 2;
	}
	else{
	
	arr[i] =arr[i-1]+count;
	count = count +1;
	}
	System.out.print(arr[i]+" ");
	}
	}
	}	

DAY 3 CRITERIA	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	
	Scanner sc = new Scanner(System.in);
	int n = sc.nextInt();
	sc.close();
	
	int count = 1;
	int m = 4;
	int arr[] = new int[n];	

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

          if(i==0){
             arr[i] = 5;
          }
          else{

             arr[i] =arr[i-1]+(m*count);
             count = count +1;
          }
           System.out.print(arr[i]+" ");
        }
    }
}
```

Topic OOPs, Classes and Methods

I-Practice (Mandatory)

Question	Solution	Remarks
DISPLAY ITEM TYPE	**import java.io.*;** **import java.util.*;** **class Main{** **public static void main(String[] args) throws Exception{** **Scanner sc = new Scanner(System.in);** **System.out.println("Enter the item type name");** **String name = sc.next();** **System.out.println("Enter the cost per day");** **double cost = sc.nextDouble();** **System.out.println("Enter the deposit");** **double deposite = sc.nextDouble();** **sc.close();** **ItemType i = new ItemType(name,cost,deposite);** **i.display();** **}** **}** **import java.text.*;** **public class ItemType {** **String name;** **double costPerDay;** **double deposit;** **public void display(){** **System.out.println("Item type details");** **/* System.out.println("Name : "+getName());** **System.out.println("CostPerDay : "+String.format("%.2f",getCostPerDay()));** **System.out.println("Deposit : "+String.format("%.2f",getDeposit()));*/** **System.out.printf("Name : %s\nCostPerDay : %.2f\nDeposit : %.2f",name,costPerDay,deposit);** **}** **public ItemType() {** **super();** **}** **public ItemType(String name, double costPerDay, double deposit) {** **super();** **this.name = name;** **this.costPerDay = costPerDay;** **this.deposit = deposit;** **}** **public String getName() {** **return name;** **}** **public void setName(String name) {** **this.name = name;** **}** **public double getCostPerDay() {** **return costPerDay;** **}** **public void setCostPerDay(double costPerDay) {** **this.costPerDay = costPerDay;** **}** **public double getDeposit() {** **return deposit;** **}** **public void setDeposit(double deposit) {** **this.deposit = deposit;** **}** **}**	**80% test cases are accepted**
COMPARE PHONE NUMBER - JAVA	**import java.io.*;** **import java.util.*;** **class Main{** **public static void main(String[] args) throws Exception{** **Scanner sc = new Scanner(System.in);** **System.out.println("Enter Name");** **String name = sc.next();** **System.out.println("Enter UserName");** **String userName = sc.next();** **System.out.println("Enter Password");** **String password = sc.next();** **System.out.println("Enter PhoneNo");** **long phoneNo = sc.nextLong();** **System.out.println("Enter Name");** **String name1 = sc.next();** **System.out.println("Enter UserName");** **String userName1 = sc.next();** **System.out.println("Enter Password");** **String password1 = sc.next();** **System.out.println("Enter PhoneNo");** **long phoneNo1 = sc.nextLong();** **User user1 = new User(name,userName,password,phoneNo);** **User user2 = new User(name1,userName1,password1,phoneNo1);** **if (user1.comparePhoneNumber(user2)==true) {** **System.out.println("Same Users");** **}** **else{** **System.out.println("Different Users");** **}** **}** **}** **public class User {** **private String name;** **private String username;** **private String password;** **private long phoneNo;** **public boolean comparePhoneNumber(User user) {** **if (this.phoneNo==user.phoneNo){** **return true;** **}** **else {** **return false;** **}** **}** **public User() {** **super();** **// TODO Auto-generated constructor stub** **}** **public User(String name, String username, String password, long phoneNo) {** **super();** **this.name = name;** **this.username = username;** **this.password = password;** **this.phoneNo = phoneNo;** **}** **public String getName() {** **return name;** **}** **public void setName(String name) {** **this.name = name;** **}** **public String getUsername() {** **return username;**	

		}
		public void setUserName(String username) {
		this.username = username;
		}
		
		public String getPassword() {
		return password;
		}
		public void setPassword(String password) {
		this.password = password;
		}
		public long getPhoneNo() {
		return phoneNo;
		}
		public void setPhoneNo(long phoneNo) {
		this.phoneNo = phoneNo;
		}
		}	

RECTANGLE DIMENSION CHANGE - INSTANCEOF OPERATOR	import java.util.*;
	public class Main {
	
	public static void main(String[] args) {
	
	Rectangle R1;
	
	Scanner sc = new Scanner(System.in);
	
	System.out.println("Enter the length of the rectangle");
	int len = sc.nextInt();
	
	System.out.println("Enter the width of the rectangle");
	int wi = sc.nextInt();
	
	Rectangle R = new Rectangle(len,wi);
	R.display();
	
	int area =R.area();
	System.out.println("Area of the Rectangle:"+area);
	System.out.println("Enter the new dimension");
	int d = sc.nextInt();
	sc.close();
	R1=R.dimensionChange(d);
	
	if(R1 instanceof Rectangle){}
	
	R1.display();
	area=R1.area();
	System.out.println("Area of the Rectangle:"+area);
	
	}
	
	}
		public class Rectangle{
		
		private int length;
		private int width;
		
		public Rectangle(int length, int width) {
		super();
		this.length = length;
		this.width = width;
		}
		public int getLength() {
		return length;
		}
		public void setLength(int length) {
		this.length = length;
		}
		public int getWidth() {
		return width;
		}
		public void setWidth(int width) {
		this.width = width;
		}
		
		public int area(){
		int area = getWidth() *getLength();
		
		return area;
		}
		
		void display() {
		System.out.println("Rectangle Dimension");
		System.out.println("Length:"+getLength());
		System.out.println("Width:"+getWidth());
		}
		
		Rectangle dimensionChange(int d) {
		/*int w = d*width;
		int l = d* length;
		
		Rectangle R =new Rectangle(l,w);
		R.display();
		int area2 = R.area();
		System.out.println("Area of the Rectangle:"+area2);
		if(R instanceof Rectangle){}
		return R;*/
		int length=d*getLength();
		setLength(length);
		int width=d*getWidth();
		setWidth(width);
		return new Rectangle(length, width);
		}
		
		}	

F2P4 - SIMPLIFIED FRACTION	import java.util.*;
	public class Main{
	public static void main(String args[]){
	
	Scanner sc = new Scanner(System.in);
	int num = sc.nextInt();
	int den = sc.nextInt();
	
	Fraction f = new Fraction()
	Fraction.printValue (num,den);
	}
	}
		public class Fraction{
		
		void printValue(int num, int den) {
		if(num==0){
		System.out.println(num);
		}
		else if(num/den==0) {
		System.out.println(asFraction(num,den));
		}
		else if(num%den==0) {
		System.out.println(num/den);
		}
		else {
		int mod = num%den;
		System.out.println(num/den+" "+asFraction(mod,den));
		}
		}
		public int gcd(int a, int b) {
		return b == 0 ? a : gcd(b, a % b);
		}
		
		public String asFraction(int a, int b) {
		int gcd = gcd(a, b);
		return (a / gcd) + "/" + (b / gcd);
		}
		
		}	

ARRAY - FOR EACH LOOP	import java.io.*;
	import java.util.*;
	public class Main{
	public static void main (String[] args) throws Exception{
	
	Scanner sc = new Scanner(System.in);
	
	System.out.println("Enter n :");		

	int n = sc.nextInt();
	
	int a[] = new int[n];
	String s[] = new String[n];
	
	System.out.println("Enter numbers : ");
	for(int i=0;i<n;i++){
	
	a[i] = sc.nextInt();
	
	}
	
	System.out.println("Enter strings : ");
	for(int j=0;j<n;j++){
	
	s[j] = sc.next();
	
	}
	
	sc.close();
	
	System.out.println("Displaying numbers");
	for(int number:a){
	System.out.println(number);
	
	}
	
	System.out.println("Displaying strings");
	for(String word:s){
	System.out.println(word);
	
	}
	
	}
	}
	}	

COMMAND LINE ARGUMENT - COUNT	public class Main{
	public static void main(String[] args){
	
	System.out.println("Arguments :");
	for(int i=0;i<args.length;i++) {
	
	System.out.println(args[i]);
	
	}
	System.out.println("Number of arguments is "+args.length);
	}
	}	

I-Practice (Bridges)		
Question	Solution	Remarks

CUSTOMER CLASS WITH CONSTRUCTOR	import java.io.*;
	import java.util.Scanner;
	public class Main {
	public static void main(String[] args) throws Exception, IOException {
	
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the Customer Details");
	System.out.println("Enter the name");
	String name = sc.next();
	
	System.out.println("Enter the email");
	String email = sc.next();
	
	System.out.println("Enter the type");
	String type = sc.next();
	
	System.out.println("Enter the location");
	String address = sc.next();
	
	sc.close();
	
	Customer c = new Customer (name, email, type, address);
	c.displayDetails();
	
	}
	}
	
	public class Customer{
	private String customerName;
	private String customerEmail;
	private String CustomerType;
	private String CustomerAddress;
	
	public void displayDetails() {
	System.out.println("Name: "+customerName);
	System.out.println("E-mail: "+customerEmail);
	System.out.println("Type: "+CustomerType);
	System.out.println("Location: "+CustomerAddress);
	
	}
	public String getCustomerName() {
	return customerName;
	}
	public void setCustomerName(String customerName) {
	this.customerName = customerName;
	}
	public String getCustomerEmail() {
	return customerEmail;
	}
	public void setCustomerEmail(String customerEmail) {
	this.customerEmail = customerEmail;
	}
	public String getCustomerType() {
	return CustomerType;
	}
	public void setCustomerType(String customerType) {
	CustomerType = customerType;
	}
	public String getCustomerAddress() {
	return CustomerAddress;
	}
	public void setCustomerAddress(String customerAddress) {
	CustomerAddress = customerAddress;
	}
	public Customer(String customerName, String customerEmail, String customerType, String customerAddress) {
	super();
	this.customerName = customerName;
	this.customerEmail = customerEmail;
	CustomerType = customerType;
	CustomerAddress = customerAddress;
	}
	
	}	

PAYMENT-METHOD OVERLOADING - JAVA	import java.io.*;
	import java.util.Scanner;
	public class Main {
	public static void main(String[] args) throws Exception, IOException{
		import java.text.*;
		
		public class TicketBooking{

	System.out.println("Enter the mode of Payment:\n1.Cash Payment\n2.Wallet Payment\n3.Credit Card");
	Scanner sc = new Scanner(System.in);
	int mode = sc.nextInt();
	TicketBooking T = new TicketBooking();
	switch (mode){
	case 1:
	System.out.println("Enter the Amount of Payment:");
	double amount = sc.nextDouble();
	
	T.MakePayment(amount);
	break;
	
	case 2:
	System.out.println("Enter the Wallet Number:");
	String walletno = sc.next();
	
	System.out.println("Enter the Amount of Payment:");
	double amount1 = sc.nextDouble();
	
	T.MakePayment(walletno, amount1);
	break;
	
	case 3:
	System.out.println("Enter the Credit Card Number:");
	String CCN = sc.next();
	sc.nextLine();
	
	System.out.println("Enter the Validity Date(dd/MM/yyyy):");
	String date = sc.next();
	sc.nextLine();
	
	System.out.println("Enter the Card Holder Name:");
	String name = sc.nextLine();
	
	System.out.println("Enter the Amount of Payment:");
	double amount2 = sc.nextDouble();
	
	T.MakePayment(CCN, date, name, amount2);
	break;
	
	case 4:
	System.out.println("Please select the correct mode of payment...");
	
	break;
	
	}
	
	}
	}
		public void MakePayment(double amount){
		int Amount = (int)amount;
		System.out.println("You have selected the Cash payment mode");
		System.out.println("The Amount is Rs."+Amount);
		
		}
		
		public void MakePayment(String walletNumber, double amount){
		int Amount = (int)amount;
		System.out.println("You have selected the Wallet payment mode");
		System.out.println("Wallet Number: "+walletNumber);
		System.out.println("The Amount is Rs."+Amount);
		
		}
		
		public void MakePayment(String creditCard, String ccv, String name, double amount){
		int Amount = (int)amount;
		System.out.println("You have selected the Credit Card payment mode");
		System.out.println("CreditCard Number: "+creditCard);
		System.out.println("Validity Date: "+ccv);
		System.out.println("Card Holder Name: "+name);
		System.out.println("The Amount is Rs."+Amount);
		
		}
		}	

SUM OF AN ARRAY	import java.io.*;
	import java.util.*;
	public class Main{
	public static void main (String[] args) throws Exception{
	
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter n :");
	int n = sc.nextInt();
	
	int arr[] = new int[n];
	int sum = 0;
	
	for(int i=0;i<n;i++){
	
	arr[i] = sc.nextInt();
	
	}
	
	sc.close();
	
	for(int number:arr){
	sum += number;
	
	}
	
	System.out.println("Sum of array elements is : "+ sum);
	}
	}		

COMMAND LINE ARGUMENT - PRINT STRING	import java.util.Scanner;
	class Main {
	public static void main(String[] args) {
	
	System.out.println(args[0]+" - Command Line Arguments");
	
	}
	}		

I-Practice (Hots)			
Question	Solution		Remarks

CUSTOMER ADDRESS	import java.util.Scanner;
	public class Main{
	public static void main(String[] args){
	
	System.out.println("Enter Customer Address");
	Scanner sc = new Scanner(System.in);
		public class Address{
		
		String street;
		String city;
		int pincode;
		String country;	

	System.out.println("Enter the street");
	String street = sc.nextLine();
	
	System.out.println("Enter the city");
	String city = sc.nextLine();
	
	System.out.println("Enter the pincode");
	int pincode = sc.nextInt();
	
	System.out.println("Enter the country");
	String country = sc.next();
	
	Address a = new Address(street,city,pincode,country);
	
	a.displayAddress();
	}
	}
		public Address() {
		super();
		
		}
		public Address(String street, String city, int pincode, String country) {
		super();
		this.street = street;
		this.city = city;
		this.pincode = pincode;
		this.country = country;
		}
		void displayAddress() {
		
		System.out.println("Street: "+street);
		System.out.println("City: "+city);
		System.out.println("Pincode: "+pincode);
		System.out.println("Country: "+country);
		}
		
		public String getStreet() {
		return street;
		}
		
		public void setStreet(String street) {
		this.street = street;
		}
		
		public int getPincode() {
		return pincode;
		}
		
		public void setPincode(int pincode) {
		this.pincode = pincode;
		}
		
		public String getCity() {
		return city;
		}
		
		public void setCity(String city) {
		this.city = city;
		}
		
		public String getCountry() {
		return country;
		}
		
		public void setCountry(String country) {
		this.country = country;
		}
		}	

F1P5 - BEST MOBILE PLAN	import java.io.*;
	import java.util.Scanner;
	public class Main
	{
	public static void main(String[] args)
	{
	Scanner sc = new Scanner(System.in);
	int day = sc.nextInt();
	int night = sc.nextInt();
	int weekend = sc.nextInt();
	sc.close();
	
	BestMobilePlan.printPlanDetails(day,night,weekend);
	}
	}
		//import java.text.DecimalFormat;
		public class BestMobilePlan
		{
		public static void printPlanDetails(int day,int night,int weekend)
		{
		double costA,costB;
		if(day>250){
		
		costB =(((day -250)*45)+(35*night)+(25*weekend))/100.00;
		costA =(((day -100)*25)+(15*night)+(20*weekend))/100.00;
		
		System.out.println("Plan A costs "+costA);
		System.out.println("Plan B costs "+costB);
		
		if(costA > costB){
		System.out.println("Plan B is cheapest");
		}
		else if(costA < costB){
		System.out.println("Plan A is cheapest");
		}
		else if (costA == costB){
		System.out.println("Plan A and B are the same price");
		}
		}
		else if((day<250)&&(day>100)){
		
		costA =(((day -100)*25)+(15*night)+(20*weekend))/100.00;
		costB =((35*night)+(25*weekend))/100.00;
		
		System.out.println("Plan A costs "+costA);
		System.out.println("Plan B costs "+costB);
		
		if(costA > costB){
		System.out.println("Plan B is cheapest");
		}
		else if(costA < costB){
		System.out.println("Plan A is cheapest");
		}
		else if (costA == costB){
		System.out.println("Plan A and B are the same price");
		}
		}
		else{
		costB =((35*night)+(25*weekend))/100.00;
		costA =((15*night)+(20*weekend))/100.00;
		
		System.out.println("Plan A costs "+costA);
		System.out.println("Plan B costs "+costB);
		
		if(costA > costB){
		System.out.println("Plan B is cheapest");
		}
		else if(costA < costB){
		System.out.println("Plan A is cheapest");
		}
		else if (costA == costB){
			Only 25% of Test Cases Are accepted

		System.out.println("Plan A and B are the same price"); } } } }	

COMMAND LINE ARGUMENTS II	public class Main{
	public static void main(String[] args){
	
	int sum = 0;
	
	if(args.length == 2){
	
	for(int i=0;i<args.length;i++){
	
	sum = sum + Integer.parseInt(args[i]);
	}
	
	System.out.println("The sum of "+args[0]+" and "+args[1]+" is "+sum);
	}
	else{
	
	System.out.println("Invalid Input");
	}
	}
	}		

Topic : Packages

I-Practice (Mandatory)

Question	Solution	Remarks
	Quiz	

I-Practice (Bridges)

Question	Solution	Remarks
	Quiz	

I-Practice (Hots)

Question	Solution	Remarks
	Quiz	

Topic : Inheritence

I-Practice (Mandatory)

Question	Solution			Remarks

SINGLE INHERITANCE - HCL	import java.util.Scanner;
	
	public class Main {
	public static void main(String[] args){
	
	Scanner sc=new Scanner(System.in);
	
	System.out.println("Enter the name :");
	String name = sc.nextLine();
	
	System.out.println("Enter Date of Birth :");
	String dob = sc.nextLine();
	
	System.out.println("Enter Gender :");
	String gender = sc.nextLine();
	
	System.out.println("Enter Mobile Number :");
	String mobile = sc.nextLine();
	
	System.out.println("Enter Blood Group :");
	String blood = sc.nextLine();
	
	System.out.println("Enter Blood Bank Name :");
	String Bank = sc.nextLine();
	
	System.out.println("Enter Donor Type :");
	String type = sc.nextLine();
	
	System.out.println("Enter Donation Date :");
	String date = sc.nextLine();
	
	Donor d = new Donor(name,dob,gender,mobile,blood,Bank,type,date);
	d.displayDonationDetails();
	
	}
	}
		class Person{
		
		private String name;
		private String dateOfBirth;
		private String gender;
		private String mobileNumber;
		private String bloodGroup;
		
		Person(){
		
		}
		
		public Person(String name, String dateOfBirth, String gender, String mobileNumber, String bloodGroup)
		{
		this.name = name;
		this.dateOfBirth = dateOfBirth;
		this.gender = gender;
		this.mobileNumber = mobileNumber;
		this.bloodGroup = bloodGroup;
		}
		
		public String getName() {
		return name;
		}
		
		public void setName(String name) {
		this.name = name;
		}
		
		public String getDateOfBirth() {
		return dateOfBirth;
		}
		
		public void setDateOfBirth(String dateOfBirth) {
		this.dateOfBirth = dateOfBirth;
		}
		
		public String getGender() {
		return gender;
		}
		
		public void setGender(String gender) {
		this.gender = gender;
		}
		
		public String getMobileNumber() {
		return mobileNumber;
		}
		
		public void setMobileNumber(String mobileNumber) {
			class Donor extends Person{
			
			private String bloodBankName;
			private String donorType;
			private String donationDate;
			
			public Donor() {
			}
			
			public Donor(String name, String dateOfBirth, String gender, String mobileNumber, String bloodGroup,String bloodBankName, String donorType, String donationDate) {
			super(name, dateOfBirth, gender, mobileNumber, bloodGroup);
			this.bloodBankName = bloodBankName;
			this.donorType = donorType;
			this.donationDate = donationDate;
			}
			public String getBloodBankName() {
			return bloodBankName;
			}
			
			public void setBloodBankName(String bloodBankName) {
			this.bloodBankName = bloodBankName;
			}
			
			public String getDonorType() {
			return donorType;
			}
			
			public void setDonorType(String donorType) {
			this.donorType = donorType;
			}
			
			public String getDonationDate() {
			return donationDate;
			}
			
			public void setDonationDate(String donationDate) {
			this.donationDate = donationDate;
			}
			
			public void displayDonationDetails( ) {
			System.out.println("Donation Details :");
			
			super.displayDetails();
			System.out.println("Blood Bank Name : " + bloodBankName);
			System.out.println("Donor Type : " + donorType);
			System.out.println("Donation Date : " + donationDate);
			}
			
			}	

```
                this.mobileNumber =
mobileNumber;
        }

        public String getBloodGroup() {
                return bloodGroup;
        }

        public void setBloodGroup(String
bloodGroup) {
                this.bloodGroup =
bloodGroup;
        }

   public void displayDetails() {
      System.out.println("Name : " + name);
                System.out.println("Date
Of Birth : " + dateOfBirth);
                System.out.println("Gender
: " + gender);
                System.out.println("Mobile
Number : " + mobileNumber);
                System.out.println("Blood
Group : " + bloodGroup);
        }
}
```

CALCULATE REWARD POINTS - HCL

```
import java.io.*;
import java.util.Scanner;

public class Main {
   public static void main(String[] args){

      Scanner sc = new Scanner(System.in);
      VISACard card;
      String response;


      do{
         System.out.println("Enter the transaction detail");
         String details = sc.nextLine();

         String[] detailArr = details.split(",");

         String type = detailArr[0];
         double amount = Double.parseDouble(detailArr[1]);
         String CType = detailArr[2];

         switch(CType){
            case "VISA card":
               card = new VISACard();
               System.out.format("Total reward points earned in this transaction is %.2f%n",
card.computeRewardPoints(type,amount));
               break;

            case "HPVISA card":
               card = new HPVISACard();
               System.out.format("Total reward points earned in this transaction is %.2f%n",
card.computeRewardPoints(type,amount));
               break;

            default :
               System.out.println("Invalid data");
               break;

         }

         System.out.println("Do you want to continue?(Yes/No)");
         response = sc.nextLine();

      }while(response.equals("Yes"));


   }
}
```

```
public class VISACard {

        public double computeRewardPoints(String type, double amount)    {


      return amount* 0.01;

   }

}
```

```
public class VISACard {

        public double computeRewardPoints(String type, double amount)    {


      return amount* 0.01;

   }

}
```

GST CALCULATION

```
import java.util.Scanner;

public class Main{
   public static void main(String[] args){

      Scanner sc = new Scanner(System.in);

         System.out.println("Enter event name");
         String name = sc.nextLine();

      System.out.println("Enter the cost per day");
      double cost = sc.nextDouble();

      System.out.println("Enter the number of days");
      int noOfDays = sc.nextInt();

      System.out.println("Enter the type of event\n1.Exhibition\n2.Stage Event");

      int type = sc.nextInt();
      String ty = Integer.toString(type);

      Event ev ;
      switch(type){

         case 1:
            System.out.println("Enter the number of stalls");
            int noOfStalls = sc.nextInt();
            ev = new Exhibition(name, ty, cost, noOfDays, noOfStalls);
            ev.toString();
            break;

         case 2:
```

```
class Exhibition extends Event {

   static int gst = 5;
   public int noOfStalls;


   public Exhibition(String name, String type, double costPerDay,int noOfDays,int
noOfStalls)
   {
      super(name, type, costPerDay, noOfDays);
      this.noOfStalls =noOfStalls;
   }

   public double totalCost() {

      double totalCost = (costPerDay *noOfDays)*(100 + Exhibition.gst)/100;
      return totalCost;

   }


   public String toString() {
      System.out.println("Event Details");
      System.out.format("Name:%s%n",name);
      System.out.println("Type:Exhibition");
      System.out.format("Number of stalls:%d%n",noOfStalls);
      System.out.format("Total amount:%.2f%n",totalCost());
      return "";

   }
}
```

	System.out.println("Enter the number of seats");
	int noOfSeats = sc.nextInt();
	ev = new StageEvent(name, ty, cost, noOfDays,noOfSeats);
	ev.toString();
	break;
	
	default:
	System.out.println("Invalid input");
	
	}
	
	}
	}		

	class StageEvent extends Event{
	
	private static int gst = 15;
	private int noOfSeats;
	
	public StageEvent(){
	
	}
	
	public StageEvent(String name, String type, double costPerDay, int noOfDays, int noOfSeats)
	{
	
	super(name, type, costPerDay, noOfDays);
	this.noOfSeats = noOfSeats;
	}
	
	public double totalCost() {
	
	double totalCost = (costPerDay *noOfDays)*(100 + StageEvent.gst)/100 ;
	return totalCost;
	
	}
	
	public String toString() {
	System.out.println("Event Details");
	System.out.format("Name:%s%n",name);
	System.out.println("Type:Stage Event");
	System.out.format("Number of seats:%d%n",noOfSeats);
	System.out.format("Total amount:%.2f%n",totalCost());
	return "";
	
	}
	}
		class Event{
		protected String name;
		protected String type;
		protected double costPerDay;
		protected int noOfDays;
		
		public Event(){
		
		}
		public Event(String name, String type, double costPerDay,int noOfDays)
		{
		this.name = name;
		this.type = type;
		this.costPerDay = costPerDay;
		this.noOfDays = noOfDays;
		
		}
		}	

ABSTRACT EVENT	import java.util.Scanner;
	
	public class Main {
	public static void main(String[] args){
	
	Scanner sc=new Scanner(System.in);
	
	System.out.println("Enter your choice");
	System.out.println("1.Exhibition");
	System.out.println("2.StageEvent");
	
	String ch = sc.next();
	
	Event ev;
	switch(ch){
	
	case "1":
	System.out.println("Enter the details in CSV format");
	sc.nextLine();
	String resp1 = sc.nextLine();
	
	String[] arr1 = resp1.split(",");
	
	ev = new Exhibition(arr1[0], arr1[1], arr1[2], arr1[3], Integer.parseInt(arr1[4]),
	Double.parseDouble(arr1[5]));
	ev.toString();
	break;
	
	case "2" :
	System.out.println("Enter the details in CSV format");
	sc.nextLine();
	String resp2 = sc.nextLine();
	String[] arr2 = resp2.split(",");
	
	ev = new StageEvent(arr2[0], arr2[1], arr2[2], arr2[3], Integer.parseInt(arr2[4])
	,Double.parseDouble(arr2[5]));
	ev.toString();
	break;
	
	default:
	System.out.println("Invalid choice");
	}
	
	}
	}
		public class Exhibition extends Event{
		private Integer noOfStalls;
		private Double rentPerStall;
		
		public Exhibition(String name, String detail, String type,String organiser,Integer
		noOfStalls, Double rentPerStall){
		super(name,detail,type,organiser);
		this.noOfStalls = noOfStalls;
		this.rentPerStall =rentPerStall;
		}
		//Fill your code here
		Double calculateAmount() {
		
		return noOfStalls*rentPerStall;
		}
		
		public String toString(){
		System.out.println("Exhibition Details");
		System.out.format("Event Name:%s%n",name);
		System.out.format("Detail:%s%n",detail);
		System.out.format("Type:%s%n",type);
		System.out.format("Organiser Name:%s%n",organiser);
		System.out.format("Total Cost:%.1f%n",calculateAmount());
		return "";
		}
		
		}
		
		public class StageEvent extends Event{
		
		private Integer noOfShows;
		private Double costPerShow;	

	public abstract class Event {
	protected String name;
	protected String detail;
	protected String type;
	protected String organiser;
	
	public Event(){		

	}
	public Event(String name, String detail, String type, String organiser){
	this.name=name;
	this.detail=detail;
	this.type=type;
	this.organiser=organiser;
	}
	
	abstract Double calculateAmount();
	
	public String toString(){
	System.out.println("Stage Event Details");
	System.out.format("Event Name:%s%n",name);
	System.out.format("Detail:%s%n",detail);
	System.out.format("Type:%s%n",type);
	System.out.format("Organiser Name:%s%n",organiser);
	System.out.format("Total Cost:%.1f%n",calculateAmount());
	return "";
	}
	
	public String getName() {
	return name;
	}
	
	public void setName(String name) {
	this.name = name;
	}
	
	public String getDetail() {
	return detail;
	}
	
	public void setDetail(String detail) {
	this.detail = detail;
	}
	
	public String getType() {
	return type;
	}
	
	public void setType(String type) {
	this.type = type;
	}
	
	public String getOrganiser() {
	return organiser;
	}
	
	public void setOrganiser(String organiser) {
	this.organiser = organiser;
	}
	
	}
		public StageEvent(String name, String detail, String type,String organiser,Integer noOfShows,Double costPerShow){
		super(name,detail,type, organiser);
		this.noOfShows = noOfShows;
		this.costPerShow =costPerShow;
		}
		
		//Fill your code here
		Double calculateAmount() {
		
		return noOfShows*costPerShow;
		
		}
		
		public String toString(){
		System.out.println("Stage Event Details");
		System.out.format("Event Name:%s%n",name);
		System.out.format("Detail:%s%n",detail);
		System.out.format("Type:%s%n",type);
		System.out.format("Organiser Name:%s%n",organiser);
		System.out.format("Total Cost:%.1f%n",calculateAmount());
		return "";
		}
		
		}	

OVERRIDING-SIMPLE	import java.io.IOException;
	import java.util.Scanner;
	
	public class Main {
	public static void main(String[] args) throws IOException {
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the name of the event:");
	String name = sc.nextLine();
	
	System.out.println("Enter the detail of the event:");
	String details = sc.nextLine();
	
	System.out.println("Enter the owner name of the event:");
	String owner = sc.nextLine();
	
	System.out.println("Enter the type of the event:\n1.Exhibition\n2.StageEvent");
	String type = sc.next();
	
	Event ev ;
	
	switch(type){
	
	case "1":
	System.out.println("Enter the number of stalls:");
	int noOfStalls = sc.nextInt();
	ev = new Exhibition(name, details, owner, noOfStalls);
	System.out.format("The projected revenue of the event is %.1f%n",ev.projectedRevenue());
	
	break;
	
	case "2":
	System.out.println("Enter the number of shows:");
	int noOfShows = sc.nextInt();
	System.out.println("Enter the number of seats per show:");
	int noOfSeatsPerShow = sc.nextInt();
	
	ev = new StageEvent(name, details, owner, noOfShows, noOfSeatsPerShow);
	System.out.format("The projected revenue of the event is %.1f%n",ev.projectedRevenue());
	break;
	
	default:
	System.out.println("Invalid input");
	}
	
	}
	}
		public class StageEvent extends Event {
		
		private Integer noOfShows;
		private Integer noOfSeatsPerShow;
		
		public StageEvent()
		{}
		public StageEvent(String name, String detail, String ownerName,Integer noOfShows,Integer noOfSeatsPerShow)
		{
		super(name,detail,ownerName);
		this.noOfShows = noOfShows;
		this.noOfSeatsPerShow = noOfSeatsPerShow;
		}
		
		public Double projectedRevenue()
		{
		return noOfShows*noOfSeatsPerShow*50.0;
		}
		
		}	

	public abstract class Event {	public class Exhibition extends Event{	

```
   protected String name;
   protected String detail;
   protected String ownerName;

   public Event(){

   }
   public Event(String name, String detail, String ownerName)
   {
      this.name = name;
      this.detail = detail;
      this.ownerName = ownerName;

   }
   public abstract Double projectedRevenue();

}
```

```
   //Your code here
         private Integer noOfStalls;


   public Exhibition()
   {
   }
   public Exhibition(String name, String detail, String ownerName,Integer
noOfStalls)
   {
      super(name,  detail, ownerName);
      this.noOfStalls = noOfStalls;

   }

   public Double projectedRevenue()
   {
      return noOfStalls *10000.0;

   }



}
```

I-Practice (Bridges)

Question	Solution	Remarks

EVENT DETAILS -SIMPLE OVERRIDE

```
import java.util.Scanner;
public class Main{
   public static void main(String[] args){

      Scanner sc = new Scanner(System.in);

         System.out.println("Enter Event Name");
      String name = sc.nextLine();
      System.out.println("Enter Detail");
      String detail = sc.nextLine();
      System.out.println("Enter Organizer");
      String organizer = sc.nextLine();
      System.out.println("Select Event Type 1.Exhibition 2.StageEvent");
      int type = sc.nextInt();

      Event e;
      if (type == 1) {

                  System.out.println("Enter stall count");
                  int stall = sc.nextInt();

                  e = new Exhibition(name,detail,organizer,stall);
                  System.out.println(e.toString());
                  e.totalCredit();

            }
            else if (type == 2) {

                  System.out.println("Enter Total shows");
                  int shows = sc.nextInt();

                  System.out.println("Enter seats per show");
                  int seats = sc.nextInt();

                  e = new StageEvent(name,detail,organizer,shows,seats);
                  System.out.println(e.toString());
                  e.totalCredit();

            }

            else {
                  System.out.println("Enter valid choice");
            }

   }
}
```

```
class Event{
   private String name;
   private String detail;
   private String organizer;

   public Event() {
         // TODO Auto-generated constructor stub
         }

         public Event(String name, String detail, String organizer) {
               super();
               this.name = name;
               this.detail = detail;
               this.organizer = organizer;
         }

   public void totalCredit(){

      System.out.println("Credit Details");
   }
@Override
   public String toString() {
               return "Event Name : " + name + "\nEvent Detail : " + detail +  "\nEvent Organizer : " +
organizer;
         }


}
```

```
class Exhibition extends Event {

   private int stallCount;

   public Exhibition() {
         // TODO Auto-generated constructor stub
         }

         public Exhibition(String name, String detail, String organizer, int stallCount) {
               super(name, detail, organizer);
               this.stallCount = stallCount;
         }


   @Override
   public void totalCredit() {

      super.totalCredit();
      System.out.println("Total Credit Gained is "+stallCount*50);

   }

   @Override
   public String toString() {

      System.out.print(super.toString());
         return "\nStall Count : " + stallCount;

   }
}
```

```
class StageEvent extends Event{
   private int totalShow;
   private int seatPerShow;

         public StageEvent() {

         }

         public StageEvent(String name, String detail, String organizer,int totalShow, int seatPerShow ) {
               super(name, detail, organizer);

               this.totalShow = totalShow;
               this.seatPerShow = seatPerShow;
         }

   @Override
   public void totalCredit() {
      super.totalCredit();
      System.out.println("Total Credit Gained is " + ((totalShow* seatPerShow )* 100));

   }
   @Override
   public String toString() {
      System.out.print(super.toString());
         return "\nTotal Events : " + totalShow + "\nTotal Seats : " + seatPerShow;

   }
}
```

STUDENT-FEEDBACK(SINGLE INHERITANCE)

```
import java.io.*;
import java.util.*;
public class Main{
```

```
public class Student{
      private int id;
      private String    name;
```

```
class StudentRating extends Student{

private int id;
```

```
   public static void main(String[] args) throws Exception,
IOException{

      Scanner scan = new Scanner(System.in);

          System.out.println("Enter the student id");
      int id = scan.nextInt();

          System.out.println("Enter the student name");
      String name = scan.next();

          System.out.println("Enter the department");
      String department = scan.next();

          System.out.println("Enter the course id");
      int c_id = scan.nextInt();

          System.out.println("Enter the Rating id");
      int r_id = scan.nextInt();

          System.out.println("Enter review");
      scan.nextLine();
      String review = scan.nextLine();

          System.out.println("Enter number of stars");
      int stars = scan.nextInt();

      Student s1 = new
StudentRating(id,name,department,c_id,r_id,review,stars);
          System.out.println(s1.toString());



   }
}
```

```
          private String    department;
          private int courseId;

public Student() {
          super();
                    // TODO Auto-generated
constructor stub
          }

          public Student(int id, String name,
String department, int courseId) {
                    super();
                    this.id = id;
                    this.name = name;
                    this.department =
department;
                    this.courseId = courseId;
          }

          @Override
          public String toString() {
                    return "Student :\nId : " + id +
"\nName : " + name + "\nDepartment : " +
department + "\nCourse Id : " + courseId;
          }



}
```

```
private String    review;
private int stars;
private int rating;
Student student;

   public StudentRating() {
          // TODO Auto-generated constructor stub
          }

          public StudentRating(int id, String name, String department, int courseId,int
id2, String review, int stars) {
                    super(id, name, department, courseId);
                    this.id = id2;
                    this.review = review;
                    this.stars = stars;



          }

          public int getId() {
                    return id;
          }

          public void setId(int id) {
                    this.id = id;
          }

          public String getReview() {
                    return review;
          }

          public void setReview(String review) {
                    this.review = review;
          }

          public int getStars() {
                    return stars;
          }

          public void setStars(int stars) {
                    this.stars = stars;
          }

          public int getRating() {
                    return rating;
          }

          public void setRating(int rating) {
                    this.rating = rating;
          }

public String toString(){

          System.out.println(super.toString());
          return "Rating ID : " + id + "\nReview : " + review + "\nRating Stars : " + stars;


}


}
```

OVERLOADING MAKEPAYMENT()

```
import java.util.*;

public class Main {

          public static void main(String[] args){

      String stageEvent;
      String customer;
                    String noOfSeats_s;
                    int noOfSeats;

                    Scanner scan = new Scanner(System.in);

                    System.out.println("Enter the Booking details");
                    String csv = scan.nextLine();


                    String[] stringarray = csv.split(",");
                    stageEvent = stringarray[0];
                    customer = stringarray[1];
                    noOfSeats_s = stringarray[2];
                    noOfSeats=Integer.parseInt(noOfSeats_s);

                    System.out.println("Payment mode\n"+ "1.Cash payment\n"+
"2.Wallet payment\n"+ "3.Credit card payment");
                    int option = scan.nextInt();
                    TicketBooking tb = new
TicketBooking(stageEvent,customer,noOfSeats);
                    double amount;

                    if(option==1) {
                              System.out.println("Enter the amount");
                              amount = scan.nextDouble();
                    System.out.println(tb.toString());
                    tb.makePayment(amount);

                    } else if(option==2) {
                              System.out.println("Enter the amount");
                              amount = scan.nextDouble();
                              System.out.println("Enter the wallet number");
                              scan.nextLine();
                              String walletNumber = scan.nextLine();


                    System.out.println(tb.toString());
                    tb.makePayment(walletNumber,amount);
```

```


public class TicketBooking {
          private String stageEvent;
   private String customer;
   private int noOfSeats;

          public TicketBooking() {
                    super();
                    // TODO Auto-generated constructor stub
          }

          public TicketBooking(String stageEvent, String customer, int noOfSeats) {
                    this.stageEvent = stageEvent;
                    this.customer = customer;
                    this.noOfSeats = noOfSeats;
          }

          public String getStageEvent() {
                    return stageEvent;
          }

          public void setStageEvent(String stageEvent) {
                    this.stageEvent = stageEvent;
          }

          public String getCustomer() {
                    return customer;
          }

          public void setCustomer(String customer) {
                    this.customer = customer;
          }

          public int getNoOfSeats() {
                    return noOfSeats;
          }

          public void setNoOfSeats(int noOfSeats) {
                    this.noOfSeats = noOfSeats;
          }

   public void makePayment(Double amount) {
          System.out.println("Amount " + amount +" paid in cash");
   }
```

```
                }else if(option==3) {
                   System.out.println("Enter card holder name");
                scan.nextLine();
                        String name= scan.nextLine();

                        System.out.println("Enter the amount");
                        amount = scan.nextDouble();


                        System.out.println("Enter the credit card type");
                        scan.nextLine();
                        String c_type = scan.nextLine();

                        System.out.println("Enter the CCV number");
                        String ccv = scan.nextLine();

                 System.out.println(tb.toString());
                        tb.makePayment(c_type,ccv,name,amount);

                }else {
                        System.out.println("Invalid choice");
                }

        }

}
```

```
   public void makePayment(String walletNumber ,Double amount) {
        System.out.println("Amount " + amount +" paid using wallet number " + walletNumber);
   }
   public void makePayment(String creditCard,String ccv,String name,Double amount) {
      System.out.println("Holder name:" + name + "\nAmount "+ amount +" paid using " + creditCard +" card"
                        + "\nCCV:" +ccv);
   }


  // @Override
        public String toString() {
                return "Stage event:" + stageEvent + "\nCustomer:" + customer + "\nNumber of seats:" +
noOfSeats;
        }


}
```

I-Practice (Hots)

Question	Solution		Remarks

ACCOUNT DETAILS - HCL	import java.io.*;
	import java.util.Scanner;
	
	public class Main {
	  public static void main(String[] args){
	
	  System.out.println("Enter User Details(HolderName,Account Number,IFSC code,Contact Number)");
	
	  Scanner sc = new Scanner(System.in);
	
	  String details = sc.nextLine();
	  String[] arr = details.split(",");
	
	  String holderName = arr[0];
	  long accNo =Long.parseLong(arr[1]);
	  String IFSC = arr[2];
	  long ContactNo = Long.parseLong(arr[3]);
	
	  System.out.println("Enter Account Type");
	  String Type = sc.nextLine();
	
	  Account acc;
	  if(Type.equalsIgnoreCase("Current")) {
	
	  System.out.println("Enter organization Name");
	  String orgName = sc.nextLine();
	
	  System.out.println("Enter TIN number");
	  long TIN = sc.nextLong();
	
	  acc = new CurrentAccount(holderName, accNo, IFSC, ContactNo, orgName, TIN);
	  acc.display();
	
	  }
	
	  else if(Type.equalsIgnoreCase("Saving")){
	
	  System.out.println("Enter Interest Rate");
	  double interestRate = sc.nextDouble();
	
	  acc = new SavingAccount(holderName, accNo, IFSC, ContactNo, interestRate);
	  acc.display();
	
	  }
	
	  else{
	
	  System.out.println("Enter valid Account Type");
	  System.exit(0);
	  }
	  }
	}
		class Account{
		
		  private String holderName;
		  private long accNumber;
		  private String IFSCCode;
		  private long contactNumber;
		
		  public void display() {
		
		  System.out.println("Your Contact Details");
		  System.out.println("HolderName : "+holderName);
		  System.out.println("Account Number : "+accNumber);
		  System.out.println("IFSCCode : "+IFSCCode);
		  System.out.println("ContactNumber : "+contactNumber);
		
		  }
		
		  public String getHolderName() {
		  return holderName;
		  }
		
		  public void setHolderName(String holderName) {
		  this.holderName = holderName;
		  }
		
		  public long getAccNumber() {
		  return accNumber;
		  }
		
		  public void setAccNumber(long accNumber) {
		  this.accNumber = accNumber;
		  }
		
		  public String getIFSCCode() {
		  return IFSCCode;
		  }
		
		  public void setIFSCCode(String iFSCCode) {
		  IFSCCode = iFSCCode;
		  }
		
		  public long getContactNumber() {
		  return contactNumber;
		  }
		
		  public void setContactNumber(long contactNumber) {
		  this.contactNumber = contactNumber;
		  }
		
		  public Account(String holderName, long accNumber, String iFSCCode, long contactNumber) {
		  super();
		  this.holderName = holderName;
		  this.accNumber = accNumber;
		  IFSCCode = iFSCCode;
		  this.contactNumber = contactNumber;
		  }
		
		}
		
		class SavingAccount extends Account{
		
		private double interestRate;
		
		  public SavingAccount(String holderName, long accNumber, String iFSCCode, long contactNumber, double interestRate) {
		  super(holderName, accNumber, iFSCCode, contactNumber);
		  this.interestRate = interestRate;
		  }
		
		  public void display() {
		
		  super.display();
		
		  if(interestRate % 1 == 0){
		  int x = (int)Math.round(interestRate);
		  System.out.printf("Interest Rate : "+ x);
		  }
		
		public class CurrentAccount extends Account{
		private String orgName;
		  private long TIN;
		
		  //@Override
		  public void display() {
		
		  super.display();
		  System.out.println("Organization Name : "+orgName);
		  System.out.println("TIN : "+TIN);
		
		  }
		
		  public String getOrgName() {
		  return orgName;
		  }
			Only 80% Test Cases Are Accepted

	}
	
	public SavingAccount(String holderName, long accNumber, String iFSCCode, long contactNumber) {
	super(holderName, accNumber, iFSCCode, contactNumber);
	// TODO Auto-generated constructor stub
	}
	
	public double getInterestRate() {
	return interestRate;
	}
	
	public void setInterestRate(double interestRate) {
	this.interestRate = interestRate;
	}
	
	}
		public void setOrgName(String orgName) {
		this.orgName = orgName;
		}
		
		public long getTIN() {
		return TIN;
		}
		
		public void setTIN(long tIN) {
		TIN = tIN;
		}
		
		public CurrentAccount(String holderName, long accNumber, String iFSCCode, long contactNumber) {
		super(holderName, accNumber, iFSCCode, contactNumber);
		// TODO Auto-generated constructor stub
		}
		
		public CurrentAccount(String holderName, long accNumber, String iFSCCode, long contactNumber, String orgName,
		long tIN) {
		super(holderName, accNumber, iFSCCode, contactNumber);
		this.orgName = orgName;
		TIN = tIN;
		}
		
		}	

INHERITANCE - BOOKING TICKETS	import java.io.BufferedReader;
	import java.io.IOException;
	import java.io.InputStreamReader;
	
	public class Main {
	
	public static void main(String args[]) throws IOException {
	BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
	
	System.out.println("Enter the name of the Aircraft");
	String aircraftName=br.readLine();
	System.out.println("Enter the source");
	String source=br.readLine();
	System.out.println("Enter the destination");
	String destination = br.readLine();
	System.out.println("Enter the type of Flight\n1.Public Aircraft\n2.Private Aircraft");
	int choice=Integer.parseInt(br.readLine());
	
	if(choice==1)
	{
	System.out.println("Is the flight check in before two hours");
	String ans = br.readLine();
	Boolean b1;
	if(ans.equals("yes"))
	{
	b1=true;
	}
	else
	{
	b1=false;
	}
	System.out.println("Enter the number of kgs allowed per person");
	int noOfKgsallowed=Integer.parseInt(br.readLine());
	
	System.out.println("Enter the additional fee charged for extra baggage per Kg");
	float additionalFeeperkg=Float.parseFloat(br.readLine());
	
	Aircraft b = new PublicAircraft(aircraftName,source,destination,b1,noOfKgsallowed,additionalFeeperkg);
	System.out.println("Flight Details :");
	System.out.println("Public Aircraft:");
	b.displayDetails();
	
	}
	
	if(choice==2)
	{
	System.out.println("Is the flight check in before two hours");
	Boolean ans = Boolean.parseBoolean(br.readLine());
	System.out.println("Enter the name of the pilot chose");
	String pilotPreference=br.readLine();
	System.out.println("Enter the Purpose of your flight");
	String purpose=br.readLine();
	
	Aircraft b = new PrivateAircraft(aircraftName,source,destination,ans,pilotPreference,purpose);
	System.out.println("Flight Details :");
	System.out.println("Private Aircraft:");
	b.displayDetails();
	
	}
	
	}
	}
		public class Aircraft {
		
		protected String aircraftName;
		protected String source;
		protected String destination;
		public String getAircraftName() {
		return aircraftName;
		}
		public void setAircraftName(String aircraftName) {
		this.aircraftName = aircraftName;
		}
		public String getSource() {
		return source;
		}
		public void setSource(String source) {
		this.source = source;
		}
		public String getDestination() {
		return destination;
		}
		public void setDestination(String destination) {
		this.destination = destination;
		}
		public Aircraft(String aircraftName, String source, String destination) {
		super();
		this.aircraftName = aircraftName;
		this.source = source;
		this.destination = destination;
		}
		
		public void displayDetails () {
		System.out.println("Aircraft Name : "+aircraftName+"\nSource : "+source+"\nDestination : "+destination);
		}
		
		}	

	public class PrivateAircraft extends Aircraft{
	private boolean checkinbeforetwohours;
	private String pilotPreference;
	private String purpose;
	String result;
	
	public boolean isCheckinbeforetwohours() {
	return checkinbeforetwohours;
	}
		public class PublicAircraft extends Aircraft{
		private boolean checkinbeforetwohours;
		private int noOfKgsallowed;
		private float additionalFeeperkg;
		String result;
		
		public boolean isCheckinbeforetwohours() {
		return checkinbeforetwohours;
		}	

```
	public void setCheckinbeforetwohours(boolean checkinbeforetwohours) {
		this.checkinbeforetwohours = checkinbeforetwohours;
	}
	public String getPilotPreference() {
		return pilotPreference;
	}
	public void setPilotPreference(String pilotPreference) {
		this.pilotPreference = pilotPreference;
	}
	public String getPurpose() {
		return purpose;
	}
	public void setPurpose(String purpose) {
		this.purpose = purpose;
	}
	public PrivateAircraft(String aircraftName, String source, String destination,
boolean checkinbeforetwohours,
				String pilotPreference, String purpose) {
		super(aircraftName, source, destination);
		this.checkinbeforetwohours = checkinbeforetwohours;
		this.pilotPreference = pilotPreference;
		this.purpose = purpose;
      result = checkinbeforetwohours ? "Yes" : "No";
	}

   public void displayDetails () {
		super.displayDetails();
		System.out.println("Flight check in before two hours :
"+result+"\nPilot chose : "+pilotPreference+"\nPurpose of the flight : "+purpose);
	}

}
```

```
	public void setCheckinbeforetwohours(boolean checkinbeforetwohours) {
		this.checkinbeforetwohours = checkinbeforetwohours;
	}

	public int getNoOfKgsallowed() {
		return noOfKgsallowed;
	}

	public void setNoOfKgsallowed(int noOfKgsallowed) {
		this.noOfKgsallowed = noOfKgsallowed;
	}

	public float getAdditionalFeeperkg() {
		return additionalFeeperkg;
	}

	public void setAdditionalFeeperkg(float additionalFeeperkg) {
		this.additionalFeeperkg = additionalFeeperkg;
	}

	public PublicAircraft(String aircraftName, String source, String destination, boolean
checkinbeforetwohours,
				int noOfKgsallowed, float additionalFeeperkg) {
		super(aircraftName, source, destination);
		this.checkinbeforetwohours = checkinbeforetwohours;
		this.noOfKgsallowed = noOfKgsallowed;
		this.additionalFeeperkg = additionalFeeperkg;
      result = checkinbeforetwohours ? "Yes" : "No";
	}

   public void displayDetails () {
		super.displayDetails();
		System.out.println("Flight check in before two hours : "+result+"\nNumber of kgs allowed per
person : "+noOfKgsallowed+"\nAdditional fee charged for extra baggage per Kg : "+additionalFeeperkg);
	}

}
```

ABSTRACT CLASS - INTRODUCTION

```
import java.text.DecimalFormat;
import java.util.Scanner;
public class Main {
	public static void main(String args[]) throws Exception {

      System.out.println("List of Shapes:\n1.Circle\n2.Rectangle\n3.Square");
		Scanner sc = new Scanner(System.in);

			System.out.println("Enter your choice:");
			int option = sc.nextInt();

			DecimalFormat deci = new DecimalFormat("0.00");
			Shape s;
			switch(option) {

			case 1:
				System.out.println("Enter the radius of the Circle:");
				float r = sc.nextFloat();
				s = new Circle(r);
				System.out.println("The perimeter is
"+deci.format(s.calculatePerimeter()));
				break;

			case 2:
				System.out.println("Enter the length of the Rectangle:");
				float l = sc.nextFloat();
				System.out.println("Enter the breadth of the Rectangle:");
				float b = sc.nextFloat();
				s = new Rectangle(l,b);
				System.out.println("The perimeter is
"+deci.format(s.calculatePerimeter()));
				break;

			case 3:
				System.out.println("Enter the side of the Square:");
				float side = sc.nextFloat();
				s = new Square(side);
				System.out.println("The perimeter is
"+deci.format(s.calculatePerimeter()));
				break;

			default:
				System.out.println("Invalid option");
				break;
			}
	}
}
```

```
public abstract class Shape {

	public abstract Double calculatePerimeter();

}
```

```
public class Square extends Shape{
	private float side;

	public Square(float side) {
		super();
		this.side = side;
	}

	public Square() {
		super();
		// TODO Auto-generated constructor stub
	}

	@Override
	public Double calculatePerimeter() {
		// TODO Auto-generated method stub
		return (double) (4*side);
	}

	public float getSide() {
		return side;
	}

	public void setSide(float side) {
		this.side = side;
	}
}
```

```
public class Rectangle extends Shape {
	private float length;
   private float breadth;
	public Rectangle(float length, float breadth) {
		super();
		this.length = length;
		this.breadth = breadth;
	}
	public Rectangle() {
		super();
		// TODO Auto-generated constructor stub
	}
	@Override
	public Double calculatePerimeter() {
		// TODO Auto-generated method stub
		return (double) (2*(length+breadth));
	}
	public float getLength() {
		return length;
	}
	public void setLength(float length) {
		this.length = length;
```

```
public class Circle extends Shape {

	private float radius;

	public Circle() {
		super();
		// TODO Auto-generated constructor stub
	}

	public Circle(float radius) {
		super();
		this.radius = radius;
	}
	@Override
	public Double calculatePerimeter() {
		// TODO Auto-generated method stub
		return 2*3.14*radius;
	}


	public float getRadius() {
		return radius;
```

```
        }
        public float getBreadth() {
                return breadth;
        }
        public void setBreadth(float breadth) {
                this.breadth = breadth;
        }
}
```

```
        }

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

}
```

Topic : Interface and Inner Class

I-Practice (Mandatory)

Question	Solution	Remarks

Java Interfaces

```
class Account
{
   private String name;
   private String accountNumber;
   private  double balance;
   private String startDate;

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

        public String getAccountNumber() {
                return accountNumber;
        }
        public void setAccountNumber(String accountNumber) {
                this.accountNumber = accountNumber;
        }

        public double getBalance() {
                return balance;
        }
        public void setBalance(double balance) {
                this.balance = balance;
        }

   public String getStartDate() {
                return startDate;
        }
        public void setStartDate(String startDate) {
                this.startDate = startDate;
        }

        public Account(String name, String accountNumber,
double balance, String startDate) {
                super();
                this.name = name;
                this.accountNumber = accountNumber;
                this.balance = balance;
                this.startDate = startDate;
        }



}
```

```
public interface  MaintainanceCharge {

   public float
calculateMaintancecharge(float
no_of_years);
}



class SavingsAccount implements
MaintainanceCharge
{
  //@Override
   public float
calculateMaintancecharge(float
no_of_years) {

                float charge =
(2*50*no_of_years)+50;
                return charge;
        }
}



public class CurrentAccount implements
MaintainanceCharge
{
  // @Override
   public float
calculateMaintancecharge(float
no_of_years) {

                float charge =
(100*no_of_years)+200;
                return charge;
        }


}
```

```
import java.util.*;
import java.text.DecimalFormat;



public class Main
{
   public static void main(String args[])
   {
      System.out.println("1.Current Account");
           System.out.println("2.Savings Account");

                Scanner sc =new Scanner(System.in);
                int  type = sc.nextInt();

                System.out.println("Name");
                String name =sc.next();

                System.out.println("Account Number");
                String AcNo =sc.next();

                System.out.println("Account Balance");
                double Acbalance = sc.nextDouble();

                System.out.println("Enter the Start Date(yyyy-mm-dd)");
                String date  =sc.next();

                System.out.println("Enter the Years");
                int n = sc.nextInt();

                DecimalFormat deci = new DecimalFormat("0.00");

                if (type == 1) {
                        CurrentAccount ca = new CurrentAccount();

                        System.out.println("Maintenance Charge For Current
Account \n"+deci.format(ca.calculateMaintancecharge(n)));
                }

                if (type == 2) {
                        SavingsAccount sa = new SavingsAccount();

                        System.out.println("Maintenance Charge For Savings
Account \n"+deci.format(sa.calculateMaintancecharge(n)));
                }

   }
}
```

INTERFACE

```
public interface Stall {

      public abstract void display();
}
```

```
import java.util.Scanner;
import java.io.IOException;
public class Main {
   public static void main(String[] args) throws IOException{

      String Stallname;
      Integer cost;
      String owner;
      Scanner sc = new Scanner(System.in);

                System.out.println("Choose Stall Type\n1)Gold Stall\n2)Premium Stall\n3)Executive Stall");
                String type = sc.next();


      Stall s;
                switch(type){

                case "1":
                        System.out.println("Enter Stall details in comma separated(Stall Name,Stall Cost,Owner
Name,Number of TV sets)");
                        sc.nextLine();
        String resp1 = sc.nextLine();
        String[] arr1=resp1.split(",");
        Stallname = arr1[0];
        cost = Integer.parseInt(arr1[1]);
        owner = arr1[2];
        Integer tvset = Integer.parseInt(arr1[3]);
        s= new GoldStall(Stallname,cost,owner,tvset);
        s.display();
        break;


                case "2":
                        System.out.println("Enter Stall details in comma separated(Stall Name,Stall Cost,Owner
Name,Number of Projectors)");
                sc.nextLine();
        String resp2 = sc.nextLine();
```

```
public class GoldStall implements Stall{

   private String stallName;
   private Integer cost;
        private String ownerName;
        private Integer tvSet;


        public String getStallName() {
                return stallName;
        }


        public void setStallName(String stallName) {
                this.stallName = stallName;
        }


        public Integer getCost() {
                return cost;
        }


        public void setCost(Integer cost) {
                this.cost = cost;
        }


        public String getOwnerName() {
                return ownerName;
        }


        public void setOwnerName(String ownerName) {
                this.ownerName = ownerName;
        }


        public Integer getTvSet() {
                return tvSet;
        }


        public void setTvSet(Integer tvSet) {
                this.tvSet = tvSet;
```

```
        String[] arr2 = resp2.split(",");
        Stallname = arr2[0];
        cost = Integer.parseInt(arr2[1]);
        owner = arr2[2];
        Integer projector = Integer.parseInt(arr2[3]);
        s= new PremiumStall(Stallname,cost,owner,projector);
        s.display();
        break;

      case "3":
                System.out.println("Enter Stall details in comma separated(Stall Name,Stall Cost,Owner
Name,Number of Screens)");
                sc.nextLine();
        String resp3 = sc.nextLine();
        String[] arr3 = resp3.split(",");
        Stallname = arr3[0];
        cost = Integer.parseInt(arr3[1]);
        owner = arr3[2];
        Integer screen = Integer.parseInt(arr3[3]);
        s= new ExecutiveStall(Stallname,cost,owner,screen);
        s.display();
        break;

                default:
                        System.out.println("Invalid Stall Type");
                }
          }
}
```

```
        }

   public GoldStall(){}
        public GoldStall(String stallName, Integer cost, String ownerName, Integer
tvSet) {
                super();
                this.stallName = stallName;
                this.cost = cost;
                this.ownerName = ownerName;
                this.tvSet = tvSet;
        }

   public String toString() {
                return "stallDemo [stallName=" + stallName + ", cost=" + cost + ",
ownerName=" + ownerName + ", tvSet="
                                + tvSet + "]";
        }


   @Override
        public void display() {
                System.out.println("Stall Name:"+stallName);
                System.out.println("Cost:"+cost+".Rs");
                System.out.println("Owner Name:"+ownerName);
                System.out.println("Number of TV sets:"+tvSet);

        }

}
```

```
public class PremiumStall implements Stall{
   private String stallName;
   private Integer cost;
   private String ownerName;
        private Integer projector;


        public String getStallName() {
                return stallName;
        }


        public void setStallName(String stallName) {
                this.stallName = stallName;
        }


        public Integer getCost() {
                return cost;
        }


        public void setCost(Integer cost) {
                this.cost = cost;
        }


        public String getOwnerName() {
                return ownerName;
        }


        public void setOwnerName(String ownerName) {
                this.ownerName = ownerName;
        }


        public Integer getprojector() {
                return projector;
        }


        public void setProjector(Integer projector) {
                this.projector = projector;
        }

   public PremiumStall(){}

        public PremiumStall(String stallName, Integer cost, String ownerName, Integer projector) {
                super();
                this.stallName = stallName;
                this.cost = cost;
                this.ownerName = ownerName;
                this.projector = projector;
        }

   public String toString() {
                return "stallDemo [stallName=" + stallName + ", cost=" + cost + ", ownerName=" + ownerName +
", projector="
                                + projector + "]";
        }
   //@Override
        public void display() {
                System.out.println("Stall Name:"+stallName);
                System.out.println("Cost:"+cost+".Rs");
                System.out.println("Owner Name:"+ownerName);
                System.out.println("Number of Projectors:"+projector);


        }
}
```

```
public class ExecutiveStall implements Stall{
        private String stallName;
   private Integer cost;
   private String ownerName;
   private Integer screen;


        public String getStallName() {
                return stallName;
        }


        public void setStallName(String stallName) {
                this.stallName = stallName;
        }


        public Integer getCost() {
                return cost;
        }


        public void setCost(Integer cost) {
                this.cost = cost;
        }


        public String getOwnerName() {
                return ownerName;
        }


        public void setOwnerName(String ownerName) {
                this.ownerName = ownerName;
        }


        public Integer getscreen() {
                return screen;
        }


        public void setScreen(Integer screen) {
                this.screen = screen;
        }

   public ExecutiveStall(){}
        public ExecutiveStall(String stallName, Integer cost, String ownerName,
Integer screen) {
                super();
                this.stallName = stallName;
                this.cost = cost;
                this.ownerName = ownerName;
                this.screen = screen;
        }

   public String toString() {
                return "stallDemo [stallName=" + stallName + ", cost=" + cost + ",
ownerName=" + ownerName + ", screen="
                                + screen + "]";
        }
  // @Override
        public void display() {
                System.out.println("Stall Name:"+stallName);
                System.out.println("Cost:"+cost+".Rs");
                System.out.println("Owner Name:"+ownerName);
                System.out.println("Number of Screens:"+screen);


        }
}
```

INTERFACE PRACTICAL PROBLEM 1

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;


public class Main {
```

```
public interface Notification{

   public void notificationBySms();
   public void notificationByEmail();
   public void notificationByCourier();
}
```

	public static void main(String[] args) throws IOException {
	
	System.out.println("Welcome to Notification Setup\nPlease select your bank:\n1)ICICI\n2)HDFC");
	BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
	
	int type = Integer.parseInt(bf.readLine());
	BankFactory bank = new BankFactory();
	switch(type) {
	
	case 1:
	System.out.println("Enter the type of Notification you want to enter\n1)SMS\n2)Mail\n3)Courier");
	String op1 = bf.readLine();
	
	switch(op1) {
	
	case "1":
	bank.getIcici().notificationBySms();
	break;
	
	case "2" :
	bank.getIcici().notificationByEmail();
	break;
	
	case "3":
	bank.getIcici().notificationByCourier();
	break;
	default:
	System.out.println("Invalid Input");
	}
	break;
	
	case 2:
	System.out.println("Enter the type of Notification you want to enter\n1)SMS\n2)Mail\n3)Courier");
	String op2 = bf.readLine();
	
	switch(op2) {
	
	case "1":
	bank.getHdfc().notificationBySms();
	break;
	
	case "2" :
	bank.getHdfc().notificationByEmail();
	break;
	
	case "3":
	bank.getHdfc().notificationByCourier();
	break;
	
	default:
	System.out.println("Invalid Input");
	
	}
	break;
	
	default:
	System.out.println("Invalid Input");
	}
	
	}
	}
		public class HDFC implements Notification {
		
		// @Override
		public void notificationBySms() {
		System.out.println("HDFC - Notification By SMS");
		
		}
		
		//@Override
		public void notificationByEmail() {
		System.out.println("HDFC - Notification By Mail");
		
		}
		
		//@Override
		public void notificationByCourier() {
		System.out.println("HDFC - Notification By Courier");
		
		}
		
		}
		
		public class ICICI implements Notification {
		
		// @Override
		public void notificationBySms() {
		System.out.println("ICICI - Notification By SMS");
		
		}
		
		// @Override
		public void notificationByEmail() {
		System.out.println("ICICI - Notification By Mail");
		
		}
		
		// @Override
		public void notificationByCourier() {
		System.out.println("ICICI - Notification By Courier");
		
		}
		
		}
		
		public class BankFactory {
		
		public ICICI getIcici() {
		ICICI icici = new ICICI();
		return icici;}
		
		public HDFC getHdfc() {
		HDFC hdfc = new HDFC();
		return hdfc;}
		
		}	

STATIC INNER CLASS	import java.io.IOException;
	import java.text.DecimalFormat;
	import java.util.Scanner;
	
	public class Main {
	public static void main(String[] args) throws IOException {
	
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the shape\r\n1.Rectangle\r\n2.Triangle");
	int ch = sc.nextInt();
	DecimalFormat df = new DecimalFormat("0.00");
	
	switch (ch) {
	
	case 1:
	System.out.println("Enter the length and breadth:");
	Shape.value1 = sc.nextDouble();
	Shape.value2 = sc.nextDouble();
	Shape.Rectangle rec = new Shape.Rectangle();
	Double Rarea = rec.computeRectangleArea();
	System.out.println("Area of rectangle is " + df.format(Rarea));
	break;
	
	case 2:
	System.out.println("Enter the base and height:");
	Shape.value1 = sc.nextDouble();
	Shape.value2 = sc.nextDouble();
	Shape.Triangle tri = new Shape.Triangle();
	Double Tarea = tri.computeTriangleArea();
	System.out.println("Area of triangle is " + df.format(Tarea));
	break;
	
	default:
	System.out.println("Invalid choice");
	break;
	
	} }
	}
		public class Shape {
		
		public static Double value1;
		public static Double value2;
		
		public static class Rectangle {
		public Double computeRectangleArea() {
		return (value1 * value2);
		}
		}
		
		public static class Triangle {
		public Double computeTriangleArea() {
		return ((0.5) * (value1 * value2));
		}
		}
		}	

I-Practice (Bridges)

Question	Solution		Remarks

MULTIPLE INHERITANCE WITH INTERFACES	public class Bike implements BikeDistance, BikeSpeed {
	
	private int distance;
	private int speed;
	
	public Bike() {
	super();
	
	}
	
	public Bike(int distance, int speed) {
	super();
	this.distance = distance;
	this.speed = speed;
	}
	
	public int getDistance() {
	return distance;
	}
	
	public void setDistance(int distance) {
	this.distance = distance;
	}
	
	public int getSpeed() {
	return speed;
	}
	
	public void setSpeed(int speed) {
	this.speed = speed;
	}
	public int totalDistance() {
	
	int total = 0;
	total = distance *speed;
	return total;
	
	}
	
	public int averageSpeed() {
	
	int average = 0;
	average = (distance*speed)/speed;
	return average;
	
	}
	}
		public interface BikeSpeed{
		
		public abstract int averageSpeed();
		
		}
		
		public interface BikeDistance{
		
		public abstract int totalDistance();
		
		}
		
		import java.io.*;
		import java.util.Scanner;
		
		public class Main{
		public static void main(String[] args) throws Exception, IOException{
		
		Scanner sc = new Scanner(System.in);
		
		System.out.println("Enter the distance travelled : ");
		int distance = sc.nextInt();
		
		System.out.println("Enter the speed of the vehicle : ");
		int speed = sc.nextInt();
		
		sc.close();
		
		Bike b = new Bike(distance,speed);
		
		int avg = b.averageSpeed();
		int tot = b.totalDistance();
		System.out.println("Total distance travelled : "+tot);
		System.out.println("Average speed maintained : "+avg);
		}
		}	

I-Practice (Hots)

Question	Solution		Remarks

ROAD FIGHTER(INTERFACE)	interface GameStatus {
	public void status( );
	}
	
	import java.util.Scanner;
	
	class Hero implements GameStatus{
	private int life ;
	
	public Hero(int life) {
	super();
	this.life = life;
	}
	@Override
	public void status() {
	life = life -1;
	System.out.println("You have "+life+" lives left");
	
	}
	public int getLife() {
	return life;
	}
	public void setLife(int life) {
	this.life = life;
	}
	
	}
	
	class Villan implements GameStatus{
	
	private int damage;
	public Villan(int damage) {
	super();
	this.damage = damage;
	}
	@Override
	public void status() {
	damage = damage+50;
	System.out.println("The Villain Damage level is :"+damage);
	
	}
	public int getDamage() {
	return damage;
	}
	public void setDamage(int damage) {
	this.damage = damage;
	}}
		public class Main {
		
		public static void main(String[] args) {
		
		int life = 5;
		int Damagelevel = 0;
		
		Villan v = new Villan(Damagelevel);
		Hero h = new Hero(life);
		Scanner sc = new Scanner(System.in);
		
		while((life != 0)&&(Damagelevel !=100)){
		
		System.out.println("Enter who punches :");
		System.out.println("1.Hero");
		System.out.println("2.Villain");
		
		int person = sc.nextInt();
		
		if(person== 2) {
		life = life-1;
		h.status();
		
		}
		else {
		v.status();
		Damagelevel = Damagelevel+50;
		}
		
		}
		
		sc.close();
		if(life == 0) {
		System.out.println("You lose the GAME !!!");
		}
		
		else {
		System.out.println("----GAME OVER----\n YOU WINS!!!");
		
		}
		}
		
		}	

NESTED CLASSES	public class Stall { public String name;	import java.util.*; import java.io.*;	

	public String detail;
	public String owner;
	public int cost;
	
	public Stall(String name, String detail, String owner, int cost) {
	super();
	this.name = name;
	this.detail = detail;
	this.owner = owner;
	this.cost = cost;
	}
	
	public class GoldStall {
	public int tvSet;
	
	public GoldStall(int tvSet) {
	super();
	this.tvSet = tvSet;
	
	}
	
	public class PlatinumStall {
	public int projector;
	
	public PlatinumStall(int projector) {
	super();
	this.projector = projector;
	}
	
	public void display(){
	
	int total = cost+(projector*500)+(tvSet*100);
	System.out.println("Stall Name:"+name);
	System.out.println("Details:"+detail);
	System.out.println("Owner Name:"+owner);
	System.out.println("TV Sets:"+tvSet);
	System.out.println("Projectors:"+projector);
	System.out.println("Total Cost:" +total);
	}
	}
	}
	
	}
		public class Main {
		public static void main(String[] args) {
		
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter Stall details in comma separated(Stall Name,Stall Description,Owner Name,Stall Cost,Number of TV set,Number of Projectors)");
		String s1 = sc.nextLine();
		sc.close();
		
		String [] input = s1.split(",");
		
		String name = input[0];
		String details = input[1];
		String owner = input[2];
		int cost = Integer.parseInt(input[3]);
		int tv = Integer.parseInt(input[4]);
		int proj = Integer.parseInt(input[5]);
		
		Stall c1 =new Stall(name, details, owner, cost);
		Stall.GoldStall g = c1.new GoldStall(tv);
		Stall.GoldStall.PlatinumStall p = g.new PlatinumStall(proj);
		p.display();
		
		}
		}	

Topic : Exception Handling

I-Practice (Mandatory)

Question	Solution	Remarks

EXCEPTION INTRODUCTION	import java.util.InputMismatchException;
	import java.util.Scanner;
	
	public class Main {
	
	public static void main(String[] args) {
	
	Scanner sc = new Scanner(System.in);
	
	System.out.println("Enter an integer input");
	try {
	int a = sc.nextInt();
	System.out.println("Entered value is "+a);
	sc.close();
	}
	catch(InputMismatchException e){
	
	System.out.println(e);
	
	}
	}
	}
	}	

ARITHMETIC EXCEPTION	import java.util.Scanner;
	public class Main {
	
	public static void main(String[] args) {
	
	try {
	Scanner sc = new Scanner(System.in);
	
	System.out.println("Enter the cost of the item for n days");
	int a = sc.nextInt();
	
	System.out.println("Enter the value of n");
	int b = sc.nextInt();
	System.out.println("Cost per day of the item is "+(a/b));
	sc.close();
	}
	catch(ArithmeticException e){
	
	System.out.println(e);
	
	}
	}
	
	}	

ARRAYINDEXOUTOFBOUNDSEXCEPTION	import java.util.Scanner;
	public class Main {
	public static void main(String args[]) {
	int arr[] = new int[100];
	try {
	
	Scanner sc = new Scanner(System.in);	

```
        System.out.println("Enter the number of seats to be booked:");
        int  a = sc.nextInt();
        int arr1[] = new int[a];
        int n =0;
        for(int i =0;i<a;i++){
           System.out.println("Enter the seat number "+(i+1));
            n = sc.nextInt();
            arr1[i] = n;

           arr[n-1]= 1;


        }

        sc.close();
        System.out.println("The seats booked are:");
        for(int j =0;j<arr1.length;j++){
           System.out.println(arr1[j]);

        }


    }
    catch(ArrayIndexOutOfBoundsException e){

       System.out.println(e);

    }
        }
}
```

PARSE EXCEPTION

```
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;
import java.io.*;


public class Main {

   public static void main(String[] args) {

      SimpleDateFormat d1 = new SimpleDateFormat("dd-MM-yyyy-HH:mm:ss");
         Scanner sc = new Scanner(System.in);

                  System.out.println("Enter the stage event start date and end date");
                  try {
                        String start = sc.nextLine();
                    Date sd = d1.parse(start);

                    String end = sc.nextLine();
                 Date ed = d1.parse(end);

          sc.close();

          Date d2=new Date();
          d2=sd;

          Date d3=new Date();
          d3=ed;

                 System.out.println("Start date:" + d1.format(d2));
                 System.out.println("End date:" + d1.format(d3));

                  }
                  catch(ParseException e) {
                           System.out.println("Input dates should be in the format 'dd-MM-yyyy-HH:mm:ss'");
                  }

          }

}
```

SEATNOTAVAILABLEEXCEPTION

```
import java.util.Scanner;
import java.io.*;

public class Main {
          public static void main(String args[])throws SeatNotAvailableException{

   Scanner sc = new Scanner(System.in);
      System.out.println("Enter the number of rows and columns of the show:");
      int n = sc.nextInt();
      int size = (n*n);
      int s;
      int a[] = new int[size];
      int mat[][] = new int[n][n];

      System.out.println("Enter the number of seats to be booked:");
      int seats = sc.nextInt();
      try {
         for(int i=0;i<seats;i++) {
            System.out.println("Enter the seat number "+(i+1));
            s = sc.nextInt();

            if(a[s]==0) {
               a[s] =1;
               int t=0;
               for(int j=0;j<n;j++) {
                  for(int k=0;k<n;k++) {
                     mat[j][k] = a[t];
                     t++;
                  }
               }
            }else {
               throw new SeatNotAvailableException("Already Booked");
            }
```

```
public class SeatNotAvailableException extends Exception{
          public SeatNotAvailableException(String s) {
      super(s);
   }
}
```

```
        }

        }catch (Exception e) {
        System.out.println(e);
      }
      finally {
        System.out.println("The seats booked are:");
        for(int i=0;i<n;i++) {
          for(int j=0;j<n;j++) {
            System.out.print(mat[i][j]+" ");
          }
          System.out.println();
        }

         }


   }
}
```

I-Practice (Bridges)

Question	Solution		Remarks

EVENTTYPEDOESNOTEXISTSEXCEPTION	`import java.util.ArrayList;`
	`import java.util.List;`
	`import java.util.Scanner;`
	
	`public class Main {`
	`public static void main(String args[])throws EventTypeDoesNotExistsException {`
	
	`ArrayList<EventType> typeList=new ArrayList<>();`
	`typeList.add(new EventType("Stage Event",new Long(1)));`
	`typeList.add(new EventType("Exhibition",new Long(2)));`
	`typeList.add(new EventType("Sports meet",new Long(3)));`
	
	`ArrayList<Event> al=new ArrayList<>();`
	`Scanner sc = new Scanner(System.in);`
	`System.out.println("Enter the number of the events:");`
	`int no = sc.nextInt();`
	`sc.nextLine();`
	
	`for(int i =0;i<no;i++) {`
	`System.out.println("Enter the details of event "+(i+1));`
	`String details = sc.nextLine();`
	`String [] arr = details.split(",");`
	`String name = arr[0];`
	`String detail = arr[1];`
	`String ownerName =arr[2];`
	`Long typeId= Long.parseLong(arr[3]);`
	`try{`
	`isValid(typeId, typeList);`
	
	`while( isValid(typeId, typeList)){`
	`System.out.println("Enter the correct event type id:");`
	`typeId= sc.nextLong();`
	`}`
	`}catch(EventTypeDoesNotExistsException e){`
	`System.out.println(e);`
	`}`
	`Event e = new Event(name,detail,ownerName,typeId);`
	`al.add(e);`
	`}`
	`System.out.println("The events entered are:");`
	`System.out.printf("%-15s%-15s%-15s%-15s","Name","Details","Owner name","Event typeid ");`
	`System.out.println();`
	`for(Event event:al) {`
	`System.out.printf("%-15s%-15s%-15s%-15s\n",event.getName(),event.getDetail(),event.getOwnerName(),event.getTypeId());`
	`}`
	
	`}`
	`public static Boolean isValid(Long typeId,List<EventType>typeList)throws EventTypeDoesNotExistsException{`
	`EventTypeDoesNotExistsException en =new EventTypeDoesNotExistsException("EventTypeDoesNotExistsException: No event type available with the given id");`
	
	`for(EventType et:typeList){`
	`if(typeId == 1||typeId == 2||typeId==3) {`
	
	`}`
	`else {`
	
	`throw en;`
	`}`
	
	`}`
	
	`return false;`
	`}`
	`}`
		`public class Event {`
		
		`private String name;`
		`private String detail;`
		`private String ownerName;`
		`private Long typeId;`
		`//@Override`
		`public String toString() {`
		`return String.format("%-15s%-15s%-15s%-15s",getName(),getDetail(),getOwnerName(),getTypeId());`
		`}`
		`public Event(String name, String detail, String ownerName, Long typeId) {`
		`super();`
		`this.name = name;`
		`this.detail = detail;`
		`this.ownerName = ownerName;`
		`this.typeId = typeId;`
		`}`
		`public String getName() {`
		`return name;`
		`}`
		`public void setName(String name) {`
		`this.name = name;`
		`}`
		`public String getDetail() {`
		`return detail;`
		`}`
		`public void setDetail(String detail) {`
		`this.detail = detail;`
		`}`
		`public String getOwnerName() {`
		`return ownerName;`
		`}`
		`public void setOwnerName(String ownerName) {`
		`this.ownerName = ownerName;`
		`}`
		`public Long getTypeId() {`
		`return typeId;`
		`}`
		`public void setTypeId(Long typeId) {`
		`this.typeId = typeId;`
		`}`
		`}`
			Only 50% Test Cases Are Accepted

	`public class EventType {`
	
	`private long id;`
	`private String name;`
	
	`public long getId() {`
	`return id;`
	`}`
	
	`public void setId(long id) {`
	`this.id = id;`
	`}`
		`public class EventTypeDoesNotExistsException extends Exception{`
		
		`public EventTypeDoesNotExistsException(String message) {`
		`super(message);`
		`// TODO Auto-generated constructor stub`
		`}`
		
		`}`	

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

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

	@Override
	public String toString() {
		return "EventType [id=" + id + ", name=" + name + "]";
	}

	public EventType(String name ,long id ) {
		super();
		this.id = id;
		this.name = name;
	}

}
```

I-Practice (Hots)

Question	Solution	Remarks

INPUT MISMATCH EXCEPTION	`import java.util.*;`
	`import java.io.*;`
	`public class Main {`
	
	`public static void main(String[] args) {`
	
	`Scanner sc = new Scanner(System.in);`
	
	`try{`
	`System.out.println("Enter an integer input");`
	`int value = sc.nextInt();`
	`System.out.println("Entered value is "+value);`
	
	`}`
	`catch(InputMismatchException e){`
	`System.out.println(e);`
	
	`}`
	`}`
	`}`	

WEAK PASSWORD EXCEPTION	`import java.util.Scanner;`
	`public class Main {`
	
	`public static void main(String[] args)throws WeakPasswordException {`
	
	`Scanner sc = new Scanner(System.in);`
	`System.out.println("Enter the user details");`
	`String details = sc.nextLine();`
	`String [] arr = details.split(",");`
	
	`String name = arr[0];`
	`String mobile = arr[1];`
	`String username = arr[2];`
	`String password = arr[3];`
	`User user = new User(name,mobile,username,password);`
	`UserBO u = new UserBO();`
	`try{`
	
	`u.validate(user);`
	`}`
	`catch(WeakPasswordException e){`
	`System.out.println(e);`
	`}`
	`}`
	`}`
		`public class WeakPasswordException extends Exception{`
		
		`public WeakPasswordException(String mg) {`
		`super(mg);`
		`// TODO Auto-generated constructor stub`
		`}`
		`}`
			Only 83% Test cases are Accepted

	`public class User {`
	
	`private String name;`
	`private String mobile ;`
	`private String username;`
	`private String password;`
	
	`//  @Override`
	`public String toString() {`
	`return "Name:" + getName() + "\nMobile:" + getMobile() + "\nUsername:" + getUsername() + "\nPassword:" + getPassword() ;`
	`}`
	`public User(String name, String mobile, String username, String password) {`
	`super();`
	`this.name = name;`
	`this.mobile = mobile;`
	`this.username = username;`
	`this.password = password;`
	`}`
	`public User() {`
	`super();`
	`// TODO Auto-generated constructor stub`
	`}`
	`public String getName() {`
	`return name;`
	`}`
	`public void setName(String name) {`
	`this.name = name;`
	`}`
	`public String getMobile() {`
	`return mobile;`
	`}`
		`public class UserBO {`
		
		`public UserBO() {`
		`super();`
		`// TODO Auto-generated constructor stub`
		`}`
		`static void validate(User u) throws WeakPasswordException {`
		`String password =u.getPassword();`
		`char[] arr = password.toCharArray();`
		
		`int len = password.length();`
		
		`if(len >10 && len<=20)`
		`{`
		`int up=0, low =0, no =0, space = 0;`
		
		`for(int i =0;i<arr.length;i++) {`
		`char p = arr[i];`
		`if(Character.isUpperCase(p)) {`
		`up++;`
		`}`
		`if(Character.isLowerCase(p)) {`
		`low++;`
		`}`
		`if(Character.isDigit(p)) {`
		`no++;`
		`}`
		`if(Character.isWhitespace(p)) {`
		`space++;`
		`}`
		`}`	

```
	public void setMobile(String mobile) {
		this.mobile = mobile;
	}
	public String getUsername() {
		return username;
	}
	public void setUsername(String username) {
		this.username = username;
	}
	public String getPassword() {
		return password;
	}
	public void setPassword(String password) {
		this.password = password;
	}

}
```

```
                                int special = len - up-low-no-space;

                                if ((up>=1 ||low >=1 )&& space==0 && no >= 1 && special >=1) {

        System.out.print(u.toString());


                                }
                                else {

                                        throw  new WeakPasswordException("Your password is
weak");
                                }

                        }
                        else
                        {
                                throw  new WeakPasswordException("Your password is weak");
                        }

        }
}
```

Topic : Java Memory Management, Threads & Reflection

I-Practice (Mandatory)

Question	Solution	Remarks
ARTICLE COUNT		

```
import java.util.Scanner;
import java.io.*;

public class Main {
	public static void main(String[] args)throws Exception{

    int count = 0;
	BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
		System.out.println("Enter the number of lines");
		int n = Integer.parseInt(br.readLine());
		for(int i=1;i<=n;i++)
		{
			System.out.println("Enter line "+i);
			String line = br.readLine();
			Article obj = new Article(line);
			obj.start();
			obj.join();
			count += obj.getCount();

		}
		System.out.printf("There are %d articles in the given input\n",count);
	}
}
```

```
public class Article extends Thread{

  private String line;
  private Integer count;

	public void run() {

		String [] arr = getLine().split(" ");
		 Integer total = 0;

for(String s : arr) {
		if(s.equalsIgnoreCase("the")){
			total++;
		}
		if(s.equalsIgnoreCase("a")){
			total++;
		}
		if(s.equalsIgnoreCase("an")){
			total++;
		}
	}
	setCount(total);

    }

    public Article(String line) {
	super();
	this.line = line;
    }

    public Article() {
	super();
	// TODO Auto-generated constructor stub
    }
    public String getLine() {
	return line;
    }
    public void setLine(String line) {
	this.line = line;
    }
    public Integer getCount() {
	return count;
    }
    public void setCount(Integer count) {
	this.count = count;
    }


}
```

PROFIT OR LOSS			

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

public class Main {
	public static void main(String args[])throws NumberFormatException,
IOException {

		System.out.println("Enter the number of events");
    BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
    int noOfEvents = Integer.parseInt(in.readLine());

    if(noOfEvents < 0 || noOfEvents % 2 != 0) {
      System.out.println("Invalid Input");
      System.exit(0);
    }

    ThreadGroup group = new ThreadGroup("Events-Group");

    System.out.println("Enter event details in CSV");
    int halfPt = noOfEvents / 2;


    List <Event> events = new ArrayList<>();
    for(int i = 0; i < halfPt; ++i) {
      String [] eventDetails = in.readLine().split(",");
      String eventName = eventDetails[0];
```

```
import java.util.List;

public class ComputeStatus implements Runnable{

	private List<Event> eventList;

  public ComputeStatus(List<Event> eventList) {
		super();
		this.eventList = eventList;
	}

	public List<Event> getEventList() {
		return eventList;
	}

	public void setEventList(List<Event> eventList) {
		this.eventList = eventList;
	}
	@Override
	public void run() {

    for(Event event : eventList) {
      if(isProfitable(event)) {
        System.out.println(event.getName() + " yields profit");
      } else {
        System.out.println(event.getName() + " yields loss");
      }
    }
		}
```

```
        String hallName = eventDetails[1];
        Double cost  = Double.parseDouble(eventDetails[2]);
        Integer hallCapacity  = Integer.parseInt(eventDetails[3]);
        Integer seatsBooked = Integer.parseInt(eventDetails[4]);

        HallBooking booking = new HallBooking(hallName, cost, hallCapacity,
seatsBooked);
        Event event = new Event(eventName, booking);
        events.add(event);
      }
      Thread t1 = new Thread(group, new ComputeStatus(events));
      t1.start();
      try {
        t1.join();
      } catch (InterruptedException e) {
        e.printStackTrace();
      }

      events = new ArrayList<>();
      for(int i = halfPt; i < noOfEvents; ++i) {
        String [] eventDetails = in.readLine().split(",");
        String eventName = eventDetails[0];
        String hallName = eventDetails[1];
        Double cost  = Double.parseDouble(eventDetails[2]);
        Integer hallCapacity  = Integer.parseInt(eventDetails[3]);
        Integer seatsBooked = Integer.parseInt(eventDetails[4]);

        HallBooking booking = new HallBooking(hallName, cost, hallCapacity,
seatsBooked);
        Event event = new Event(eventName, booking);
        events.add(event);
      }
      Thread t2 = new Thread(group, new ComputeStatus(events));
      t2.start();
      try {
        t2.join();
      } catch (InterruptedException e) {
        e.printStackTrace();
      }

    }
}
```

```
  private boolean isProfitable(Event event) {
      HallBooking booking = event.getHallbooking();

      Double value = ( 100.0 * booking.getSeatsBooked())/ booking.getHallCapacity();

      return value >= 60;
    }
}
```

I-Practice (Bridges)

Question	Solution	Remarks
CITY COUNT		**Display output in incorrect manner**

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

public class Main {
        public static void main(String args[]) throws Exception {

    Scanner sc = new Scanner(System.in);
        ArrayList<User> al = new ArrayList<>();
                System.out.println("Enter the number of users");
                int n = sc.nextInt();
                sc.nextLine();

                for(int i=0;i<n;i++) {
                        System.out.println("Enter the details of user
"+(i+1));
                        String details = sc.nextLine();
                        String[] arr = details.split(",");
                        String name = arr[0];
                        String mobileNumber=arr[1];
                        String city = arr[2];
                        String state = arr[3];

                        User u1 = new
User(name,mobileNumber,city,state);
                        al.add(u1);
                }

                System.out.println("Enter the number of cities:");
                int no = sc.nextInt();
                sc.nextLine();
                 CityCount c = null;
                for(int j=0;j<no;j++) {
                        System.out.println("Enter the name of city
"+(j+1));
                        String searchcity = sc.nextLine();
          c = new CityCount(searchcity,al);
                }
      c.start();

          }
}
```

```
public class User {
        private String name;
   private String mobileNumber;
   private String city;
   private String state;

        public User() {
                super();
                // TODO Auto-generated
constructor stub
        }


        public User(String name, String
mobileNumber, String city, String state) {
                super();
                this.name = name;
                this.mobileNumber =
mobileNumber;
                this.city = city;
                this.state = state;
        }


        public String getName() {
                return name;
        }
        public void setName(String name)
{
                this.name = name;
        }
        public String getMobileNumber() {
                return mobileNumber;
        }
        public void
setMobileNumber(String mobileNumber)
{
                this.mobileNumber =
mobileNumber;
        }
        public String getCity() {
                return city;
        }
        public void setCity(String city) {
                this.city = city;
        }
        public String getState() {
                return state;
        }
        public void setState(String state) {
                this.state = state;
        }

}
```

```
import java.util.*;
import java.io.*;
import java.util.ArrayList;
import java.util.List;
public class CityCount extends Thread{

   private String city;

        private Integer count =0;
        private  List<User> userList;

   public void run() {

      Map<String, Integer> hm = new HashMap<String, Integer>();

      for (User i : userList) {
         Integer j = hm.get(i.getCity());
         hm.put(i.getCity(), (j == null) ? 1 : j + 1);
      }

      // displaying the occurrence of elements in the arraylist
      for (Map.Entry<String, Integer> val : hm.entrySet()) {
         System.out.println( val.getKey() + "--" + val.getValue());
      }
          }


        public CityCount(String city,ArrayList<User> userList) {
                super();
                this.city = city;
                this.userList = userList;
        }
        public String getCity() {
                return city;
        }
        public void setCity(String city) {
                this.city = city;
        }
        public Integer getCount() {
                return count;
        }
        public void setCount(Integer count) {
                this.count = count;
        }
        public List<User> getUserList() {
                return userList;
        }
        public void setUserList(List<User> userList) {
                this.userList = userList;
        }

}
```

I-Practice (Hots)

Question	Solution	Remarks
USER NOTIFICATION		

STALL REVENUE			
Topic : Primitive Wrapper and java.utill & java.lang classes			
I-Practice (Mandatory)			
Question	Solution		Remarks

WRAPPER CLASS – INTEGER I	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	
	System.out.println("Enter an integer");
	Scanner sc = new Scanner(System.in);
	Integer no = sc.nextInt();
	
	System.out.println("The binary equivalent of "+no+" is " +no.toBinaryString(no));
	System.out.println("The hexadecimal equivalent of "+no+" is " +no.toHexString(no));
	System.out.println("The octal equivalent of "+no+" is " +no.toOctalString(no));
	System.out.println("Byte value of "+no+" is " +no.byteValue());
	System.out.println("Short value of "+no+" is " +no.shortValue());
	System.out.println("Long value of "+no+" is " +no.longValue());
	System.out.println("Float value of "+no+" is " +no.floatValue());
	System.out.println("Double value of "+no+" is " +no.doubleValue());
	}
	}		

WRAPPER CLASS – 1	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	
	System.out.println("Maximum exponent :"+Math.getExponent(Float.MAX_VALUE));
	System.out.println("Maximum value :"+Float.MAX_VALUE);
	System.out.println("Number of bits :"+Float.SIZE);
	}
	}		

CONVERTING A STRING TO DOUBLE	import java.text.ParseException;
	import java.text.SimpleDateFormat;
	import java.util.Date;
	import java.util.Scanner;
	import java.io.*;
	
	public class Main {
	public static void main(String[] args) throws ParseException {
	
	Scanner sc = new Scanner(System.in);
	System.out.println("Enter the issue date as dd/MM/yyyy");
	String iDate = sc.nextLine();
	Date issueDay = new SimpleDateFormat("dd/MM/yyyy").parse(iDate);
	
	System.out.println("Enter the due date as dd/MM/yyyy");
	String dDate = sc.nextLine();
	Date dueDay = new SimpleDateFormat("dd/MM/yyyy").parse(dDate);
	
	System.out.println("Enter the original amount");
	String a= sc.nextLine();
	Double amount =Double.parseDouble(a);
	
	System.out.println("Enter amount paid so far");
	String aS = sc.nextLine();
	Double amountPaid =Double.parseDouble(aS);
	
	BillHeader b = new BillHeader(issueDay, dueDay, amount, amountPaid);
	System.out.println(b.toString());
	}
	}
		import java.util.Date;
		import java.text.SimpleDateFormat;
		
		public class BillHeader {
		private Date issueDate;
		private Date dueDate;
		private Double originalAmount;
		private Double amountOutstanding;
		
		SimpleDateFormat d = new SimpleDateFormat("dd/MM/YYYY");
		// @Override
		public String toString() {
		return "Issue date: " + d.format(issueDate) + "\nDue Date: " + d.format(dueDate) + "\nOriginal amount Rs." + originalAmount
		+ "\nAmount outstanding Rs." + amountOutstanding ;
		}
		public Date getIssueDate() {
		return issueDate;
		}
		public void setIssueDate(Date issueDate) {
		this.issueDate = issueDate;
		}
		public Date getDueDate() {
		return dueDate;
		}
		public void setDueDate(Date dueDate) {
		this.dueDate = dueDate;
		}
		public Double getOriginalAmount() {
		return originalAmount;
		}
		public void setOriginalAmount(Double originalAmount) {
		this.originalAmount = originalAmount;
		}
		public Double getAmountOutstanding() {
		return amountOutstanding;
		}
		public void setAmountOutstanding(Double amountOutstanding) {
		this.amountOutstanding = amountOutstanding;
		}
		public BillHeader(Date issueDate, Date dueDate, Double originalAmount, Double amountOutstanding) {
		super();
		this.issueDate = issueDate;
		this.dueDate = dueDate;
		this.originalAmount = originalAmount;
		this.amountOutstanding = amountOutstanding;
		}
		
		public BillHeader() {
		super();
		}
		}
			Only 20% Test cases Are accepted

STRING REVERSAL	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {
	System.out.println("Enter a string to reverse");
	Scanner sc = new Scanner(System.in);
	String value = sc.nextLine();
	StringBuilder revalue = new StringBuilder();
	revalue.append(value);
	System.out.println("Reverse of entered string is : "+revalue.reverse());
	}
	}		

STRING API : STARTSWITH() : ILLUSTRATION	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String[] args) throws IOException {		

```
        System.out.println("Enter the string");
        Scanner sc = new Scanner(System.in);
        String value1 = sc.nextLine();

        System.out.println("Enter the start string");
        String value2 = sc.nextLine();

        if(value1.startsWith(value2)){
           System.out.println("\""+value1+"\" starts with \""+value2+"\"");
        }
        else
        {
            System.out.println("\""+value1+"\" does not start with \""+value2+"\"");
        }
             }
}
```

STRING API : SPLIT() : ILLUSTRATION

```
import java.util.StringTokenizer;
import java.util.Scanner;
import java.io.*;
public class Main {
   public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter the string");

        String value3 = sc.nextLine();

        StringTokenizer st = new StringTokenizer(value3," ");
        StringBuffer sb = new StringBuffer();

        while(st.hasMoreElements()){

           sb.append(st.nextElement()).append(" ");
        }

        String value4 = sb.toString();
        System.out.println("The words in the string are");
        String [] arr = value4.split(" ");

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

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

        }
             }
}
```

STRING TOKENIZER

```
import java.util.Scanner;
import java.util.StringTokenizer;
import java.io.*;
public class Main {
   public static void main(String[] args) throws IOException {

      Scanner sc = new Scanner(System.in);
           String value = sc.nextLine();
      String value2 = value.replaceAll("=", " ");
      StringTokenizer st = new StringTokenizer(value2,";");

      while(st.hasMoreTokens()){

         System.out.println(st.nextToken());
      }
           }
}
```

CUSTOMER ADDRESS USING STRING BUILDER

```
import java.util.Scanner;
import java.io.*;
public class Main {
   public static void main(String[] args) throws IOException {
                System.out.println("Enter Address Details :");
          Scanner sc = new Scanner(System.in);

                System.out.println("Enter Line 1 :");
                String L1 = sc.nextLine();

                System.out.println("Enter Line 2 :");
                String L2 = sc.nextLine();

                System.out.println("Enter City :");
                String city = sc.nextLine();

                System.out.println("Enter Country :");
                String country = sc.nextLine();

                System.out.println("Enter Zip Code :");
                int zip = sc.nextInt();

          Address address = new Address(L1, L2, city, country, zip);

                address.toString();
          }
}
```

```
// fill your code here

public class Address {
          private String line1;
          private String line2;
          private String city;
          private String country;
          private int zipCode;

public Address(String line1, String line2, String city, String country, int zipCode) {
          super();
                this.line1 = line1;
                this.line2 = line2;
                this.city = city;
                this.country = country;
                this.zipCode = zipCode;
          }
          public Address() {
                super();
                // TODO Auto-generated constructor stub
          }
          @Override
          public String toString() {
                StringBuilder sb = new StringBuilder();
          StringBuilder add =sb.append("Address Details :\n" + line1 +",\n"+ line2 + ",\n" + city
+ " - "+zipCode+"\n" + country) ;
                System.out.println(add);
                return null;
          }
          public String getLine1() {
                return line1;
          }
          public void setLine1(String line1) {
                this.line1 = line1;
          }
          public String getLine2() {
                return line2;
          }
          public void setLine2(String line2) {
                this.line2 = line2;
          }
          public String getCity() {
                return city;
```

```
	}
	public void setCity(String city) {
		this.city = city;
	}
	public String getCountry() {
		return country;
	}
	public void setCountry(String country) {
		this.country = country;
	}
	public int getZipCode() {
		return zipCode;
	}
	public void setZipCode(int zipCode) {
		this.zipCode = zipCode;
	}
}
```

I-Practice (Bridges)

Question	Solution	Remarks

STRING API : ENDSWITH() : ILLUSTRATION	import java.util.Scanner;
	import java.io.*;
	public class Main {
	  public static void main(String[] args) throws IOException {
	
	  Scanner sc = new Scanner(System.in);
	  System.out.println("Enter the string");
	
	  String value1 = sc.nextLine();
	
	  System.out.println("Enter the end string");
	  String value2 = sc.nextLine();
	
	  if(value1.endsWith(value2)){
	  System.out.println("\""+value1+"\" ends with \""+value2+"\"");
	  }
	  else
	  {
	  System.out.println("\""+value1+"\" does not end with \""+value2+"\"");
	  }
	  }
	}	

WRAPPER CLASS – INTEGER II	import java.util.Scanner;
	import java.io.*;
	public class Main {
	  public static void main(String[] args) throws IOException {
	
	  Scanner sc = new Scanner(System.in);
	
	  System.out.println("Enter the binary number");
	  String binaryString = sc.nextLine();
	
	  System.out.println("Enter the octal number");
	  String octalString = sc.nextLine();
	
	  System.out.println("Enter the hexadecimal number");
	  String hexString = sc.nextLine();
	  System.out.println("The integer value of the binary number "+binaryString+" is "+Integer.parseInt(binaryString, 2));
	  System.out.println("The integer value of the octal number "+octalString+" is "+Integer.parseInt(octalString, 8));
	  System.out.println("The integer value of the hexadecimal number "+hexString+" is "+Integer.parseInt(hexString, 16));
	
	  }
	}	

STRING API : REPLACE() : ILLUSTRATION	import java.util.Scanner;
	import java.io.*;
	public class Main {
	  public static void main(String[] args) throws IOException {
	
	  System.out.println("Enter the content of the document");
	  Scanner sc = new Scanner(System.in);
	  String content = sc.nextLine();
	
	  System.out.println("Enter the old name of the company");
	  String old = sc.nextLine();
	
	  System.out.println("Enter the new name of the company");
	  String newname = sc.nextLine();
	
	  System.out.println("The content of the modified document is\n"+content.replace(old,newname));
	  }
	}	

I-Practice (Hots)

Question	Solution		Remarks

EMAIL ID COMPARISON	import java.util.*;
	import java.io.*;
	public class Main {
	  public static void main(String[] args) throws IOException {
	
	  Scanner sc = new Scanner(System.in);
	
	  System.out.println("Enter First Customer Details :");
	  System.out.println("Enter Customer Name :");
	  String name1 = sc.nextLine();
	  System.out.println("Enter Customer Email :");
	  String mail1 = sc.nextLine();
	
	  System.out.println("Enter Second Customer Details :");
	  System.out.println("Enter Customer Name :");
	  String name2 = sc.nextLine();
	  System.out.println("Enter Customer Email :");
	  String mail2 = sc.nextLine();
	
	  Customer c1 = new Customer(name1, mail1);
	  Customer c2 = new Customer(name2, mail2);
	  if(mail1.equals(mail2)) {
		// fill your code here
		
		public class Customer {
		  private String name;
		  private String email;
		
		  public Customer(String name, String email) {
		  super();
		  this.name = name;
		  this.email = email;
		  }
		
		  public String getName() {
		  return name;
		  }
		  public void setName(String name) {
		  this.name = name;
		  }
		
		  public String getEmail() {
		  return email;
		  }	

	System.out.println("The Email ids of "+c1.getName()+" and "+c2.getName()+" are equal");
	
	}
	else {
	
	System.out.println("The Email ids of "+c1.getName()+" and "+c2.getName()+" are not equal");
	
	}
	
	System.out.println("Comparing without considering the cases :");
	if(mail1.equalsIgnoreCase(mail2)) {
	
	System.out.println("The Email ids of "+c1.getName()+" and "+c2.getName()+" are Equal");
	}
	else {
	System.out.println("The Email ids of "+c1.getName()+" and "+c2.getName()+" are not equal");
	}
	
	}
	}
		public void setEmail(String email) {
		this.email = email;
		}
		}	

DATE - 2	import java.text.DateFormat;
	import java.text.ParseException;
	import java.text.SimpleDateFormat;
	import java.util.Date;
	import java.util.Scanner;
	import java.io.*;
	
	public class Main {
	public static void main(String[] args) throws IOException {
	
	try {
	Scanner sc = new Scanner(System.in);
	
	Integer day = Integer.rotateRight(sc.nextInt(),2);
	Integer month = Integer.rotateRight(sc.nextInt(),2);
	Integer year = Integer.rotateRight(sc.nextInt(),2);
	
	String date1 = year.toString()+"/"+month.toString()+"/"+day.toString();
	
	DateFormat formatter = new SimpleDateFormat("yyyy/m/d");
	Date date = formatter.parse(date1);
	System.out.println(formatter.format(date));
	
	}
	catch (ParseException e) {
	
	e.printStackTrace();
	}
	
	}
	}		

Topic : Collections and Generic			
I-Practice (Mandatory)			
Question	Solution		Remarks

ARRAYLIST - INTRODUCTION	import java.awt.List;
	import java.util.ArrayList;
	import java.util.Scanner;
	import java.io.*;
	public class Main {
	public static void main(String args[]) throws Exception{
	
	boolean result=false;
	ArrayList<String> list = new ArrayList<String>();
	int i=1;
	Scanner scanner = new Scanner(System.in);
	do{
	System.out.println("Enter the username " + i++);
	String name = scanner.nextLine();
	list.add(name);
	System.out.println("Do you want to continue?(y/n)");
	String diss = scanner.nextLine();
	result=diss.equals("y");
	}while(result);
	
	System.out.println("The Names entered are:");
	for(String s:list){
	System.out.println(s);
	}
	}
	}		

SET INTRODUCTION	import java.awt.List;
	import java.util.LinkedHashSet;
	import java.util.Scanner;
	
	public class Main {
	
	public static void main(String[] args){
	
	boolean result=false;
	LinkedHashSet<String> list = new LinkedHashSet<>();
	int i=1;
	Scanner scanner = new Scanner(System.in);
	
	do{
	System.out.println("Enter the username");
	String name = scanner.nextLine();
	list.add(name);
	System.out.println("Do you want to continue?(Y/N)");		

```
                String diss = scanner.nextLine();
                result=diss.equals("Y");
                }while(result);

                System.out.println("The unique number of usernames is "+list.size());


        }

}
```

REVERSE() METHOD

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Scanner;

public class Main {
        public static void main(String args[]) {


    Scanner sc = new Scanner(System.in);
        User user;
                ArrayList<User> hs = new ArrayList<>();
   try{
                System.out.println("Enter the number of users:");
                int no = sc.nextInt();
                sc.nextLine();

                for(int i =0;i<no;i++) {
                        System.out.println("Enter the details of User "+(i+1));
                        String details = sc.next();

                        String[] arr = details.split(",");

                        String Name = arr[0];
                        String mobileNo =arr[1];
                        String uname = arr[2];
                        String password= arr[3];

                        hs.add(new User(Name,mobileNo,uname,password));
                }
                Collections.sort(hs);
                Collections.reverse(hs);
    }catch(Exception e){}

     System.out.println("The user details in reverse order:");
                System.out.printf("%-15s %-15s\n", "Name", "Mobile number");

                for(User a:hs) {
                        System.out.printf("%-15s %-15s\n",a.getName(),a.getMobileNumber());
                }


        }
}
```

```
public class User implements Comparable {
        private String name;
  private String mobileNumber;
  private String username;
  private String password;

  @Override
  public String toString() {

                return  name +  mobileNumber ;
        }
        public User(String name, String mobileNumber, String username, String password) {
                super();
                this.name = name;
                this.mobileNumber = mobileNumber;
                this.username = username;
                this.password = password;
        }
        public User() {
                super();
                // TODO Auto-generated constructor stub
        }
        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public String getMobileNumber() {
                return mobileNumber;
        }
        public void setMobileNumber(String mobileNumber) {
                this.mobileNumber = mobileNumber;
        }
        public String getUsername() {
                return username;
        }
        public void setUsername(String username) {
                this.username = username;
        }
        public String getPassword() {
                return password;
        }
        public void setPassword(String password) {
                this.password = password;
        }
        @Override
        public int compareTo(Object o) {

                String name1=this.name;
    String name2=((User)o).name;

    return name1.compareTo(name2);

        }

}
```

GENERIC CLASSES

```
import java.util.Scanner;
import java.io.*;
public class Main {
   public static void main(String[] args) throws IOException {

        Scanner sc = new Scanner(System.in);
    Item<Integer> integeritem= new Item<Integer>();
    Item<String> stringitem = new Item<String>();

    System.out.println("Enter a integer :");
    integeritem.set(new Integer(sc.nextInt()));

    System.out.println("Enter a string :");
    sc.nextLine();
    String word = sc.nextLine();

    stringitem.set(new String(word));

    System.out.printf("Integer Value :%d\n\n", integeritem.get());
    System.out.printf("String Value :%s\n", stringitem.get());
        }
}
```

```
// fill your code here

public class Item <T> {

 private T data;

  public void set(T data) {
    this.data= data;
  }

  public T get() {
    return data;
  }

}
```

I-Practice (Bridges)

Question	Solution	Remarks

ITERATOR CLASS

```
import java.util.ArrayList;
import java.util.Iterator;
import java.util.Scanner;

public class Main {

        public static void main(String[] args){

    Scanner sc = new Scanner(System.in);
        ArrayList<Stall> al = new ArrayList<>();

                System.out.println("Enter the number of stall details");
                int no = sc.nextInt();
```

```
public class Stall {


   private String name;
   private String detail;
        private String type;
        private String ownerName;

        public String toString() {
        return String.format("%-15s %-20s %-15s %s",getName(),getDetail(),getType(),getOwnerName());
        }
```

```
                sc.nextLine();
          if(no==0){
         System.out.printf( "%-15s %-20s %-15s %s", "Name","Detail","Type","Owner
name");
          }
          else{
      for(int i=0;i<no;i++) {
                        System.out.println("Enter the stall "+(i+1)+" detail");
                        String X = sc.nextLine();
                        String [] arr= X.split(",");
                        String name = arr[0];
                        String detail= arr[1];
                        String type = arr[2];
                        String owner = arr[3];

                     Stall h = new Stall(name,detail,type,owner);
                     al.add(h);
                    }

                    Iterator<Stall> it = al.iterator();

                    while(it.hasNext()) {
                        Stall x = (Stall)it.next();
                        if(x.getName().startsWith("test"))
                            it.remove();
                    }

                    System.out.printf( "%-15s %-20s %-15s %s",
"Name","Detail","Type","Owner name");
                    System.out.println();

                    for (Stall st : al) {

                        System.out.printf( "%-15s %-20s %-15s
%s\n",st.getName(),st.getDetail(),st.getType(),st.getOwnerName());
                        System.out.printf( "%-15s %-20s %-15s
%s\n",st.getName(),st.getDetail(),st.getType(),st.getOwnerName());

      }

          }
          }

}
```

```
public Stall(String name, String detail, String type, String ownerName) {
        super();
        this.name = name;
        this.detail = detail;
        this.type = type;
        this.ownerName = ownerName;
}
public Stall() {
        super();
        // TODO Auto-generated constructor stub
}
public String getName() {
        return name;
}
public void setName(String name) {
        this.name = name;
}
public String getDetail() {
        return detail;
}
public void setDetail(String detail) {
        this.detail = detail;
}
public String getType() {
        return type;
}
public void setType(String type) {
        this.type = type;
}
public String getOwnerName() {
        return ownerName;
}
public void setOwnerName(String ownerName) {
        this.ownerName = ownerName;
}



}
```

SORT() A LIST OF OBJECTS

```
import java.util.ArrayList;
import java.util.Scanner;
import java.util.Collections;
public class Main {
        public static void main(String args[]){


    Scanner sc = new Scanner(System.in);
        ArrayList<Hall> al = new ArrayList<>();

                System.out.println("Enter the number of halls:");
                int no = sc.nextInt();

                if(no == 0){
            System.out.println("Sorted Order (from the least expensive to the
most):");
        System.out.printf( "%-15s%-15s%-15s%-15s", "Name","Contact
number","Cost per day","Owner name");
                }
    else{
       sc.nextLine();
                for(int i=0;i<no;i++) {
                        System.out.println("Enter the details of hall "+(i+1));
                        String details = sc.nextLine();
                        String [] arr= details.split(",");
                        String name = arr[0];
                        String contactno = arr[1];
                        double cost = Double.parseDouble(arr[2]);
                        String owner = arr[3];

                     Hall h = new Hall(name,contactno,cost,owner);
                     al.add(h);
                    }

    Collections.sort(al);
                System.out.println("Sorted Order (from the least expensive to the
most):");
                System.out.printf( "%-15s%-15s%-15s%-15s", "Name","Contact
number","Cost per day","Owner name");
                System.out.println();


                for (Hall hall : al) {

                        System.out.printf( "%-15s%-15s%-15s%-
15s\n",hall.getName(),hall.getContactNumber(),hall.getCostPerDay(),hall.getOwnerN
ame());

    }

                }
        }
}
```

```
public class Hall implements Comparable{

    private String name;
        private String contactNumber;
        private double costPerDay;
        private String ownerName;
//      @Override
        public int compareTo(Object o) {
                double cost1=this.costPerDay;
            double cost2=((Hall)o).costPerDay;
             if(cost1<cost2)
                return -1;
                else if(cost1>cost2)
                  return 1;
                else
                  return 0;
        }



//      @Override
        public String toString() {
                return String.format( "%-15s%-15s%-15s%-
15s",getName(),getContactNumber(),getCostPerDay(),getOwnerName());
        }



        public Hall(String name, String contactNumber, double costPerDay, String ownerName) {
                super();
                this.name = name;
                this.contactNumber = contactNumber;
                this.costPerDay = costPerDay;
                this.ownerName = ownerName;
        }



        public Hall() {
                super();
                // TODO Auto-generated constructor stub
        }

        public String getName() {
                return name;
        }
        public void setName(String name) {
                this.name = name;
        }
        public String getContactNumber() {
                return contactNumber;
        }
        public void setContactNumber(String contactNumber) {
                this.contactNumber = contactNumber;
        }
        public double getCostPerDay() {
                return costPerDay;
        }
        public void setCostPerDay(double costPerDay) {
                this.costPerDay = costPerDay;
        }
        public String getOwnerName() {
                return ownerName;
        }
        public void setOwnerName(String ownerName) {
```

<table>
<tr><td></td><td></td><td>this.ownerName = ownerName;
}

}</td><td></td></tr>
<tr><td colspan="4">I-Practice (Hots)</td></tr>
<tr><td>Question</td><td colspan="2">Solution</td><td>Remarks</td></tr>
<tr><td>LIST OF LIST</td><td colspan="2"></td><td></td></tr>
<tr><td>MIN AND MAX</td><td>import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;

public class Main {
public static void main(String[] args){

Scanner sc = new Scanner(System.in);
List <TicketBooking> al = new ArrayList<>();
System.out.println("Enter the number of customers");
int n = sc.nextInt();
if(n>0){
System.out.println("Enter the booking price accordingly with customer name in CSV(Customer Name,Price)");
sc.nextLine();

for(int i=0;i<n;i++) {
String details = sc.nextLine();
String [] arr = details.split(",");

String name = arr[0];
int price = Integer.parseInt(arr[1]);
TicketBooking t = new TicketBooking(name,price);
al.add(t);

}
Collections.sort(al);

TicketBooking x = al.get(0);
System.out.println(x.getCustomerName()+" spends minimum amount of Rs."+x.getPrice());

TicketBooking y = al.get(n-1);
System.out.println(y.getCustomerName()+" spends maximum amount of Rs."+y.getPrice());
}
else{
System.out.println("Invalid Input");
}
}
}</td><td>public class TicketBooking implements Comparable{

private String customerName;
private int price;

@Override
public int compareTo(Object o) {
int price1 = this.price;
int price2 = ((TicketBooking)o).price;

if(price1>price2)
return 1;
else if(price2>price1)
return -1;
else
return 0;
}

public String getCustomerName() {
return customerName;
}

public void setCustomerName(String customerName) {
this.customerName = customerName;
}

public int getPrice() {
return price;
}

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

public TicketBooking() {
super();
// TODO Auto-generated constructor stub
}

public TicketBooking(String customerName, int price) {
super();
this.customerName = customerName;
this.price = price;
}

}</td><td></td></tr>
<tr><td>REPLICA OF A LIST</td><td>import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Scanner;

public class Main {

public static void main(String[] args){

Scanner sc = new Scanner(System.in);
User u = null;
System.out.println("Enter number of users");
int n = sc.nextInt();
sc.nextLine();
if(n>0){
List<User> sourcelist = new ArrayList<>(n);
List<User> destinstionlist = new ArrayList<>(n);

System.out.println("Enter the user details in CSV(Username,password)");
for(int i=0;i<n;i++) {
String details = sc.nextLine();
String [] arr = details.split(",");
String username = arr[0];
String password = arr[1];
u = new User(username,password);
sourcelist.add(u);
destinstionlist.add(null);

}

u.backUp(destinstionlist, sourcelist);
System.out.println("Copy of user list:");
System.out.printf("%-20s %s\n","Username","Password");
for(User user:destinstionlist) {
System.out.printf("%-20s %s\n",user.getUsername(),user.getPassword());
}

}else{
System.out.println("Invalid Input");
}

}
}</td><td>import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class User {

private String username;
private String password;
List<User> backUp(List<User>dest , List<User> source){

Collections.copy(dest, source);
return dest;

}

}

public User(String username, String password) {
super();
this.username = username;
this.password = password;
}
public User() {
super();
// TODO Auto-generated constructor stub
}
public String getUsername() {
return username;
}
public void setUsername(String username) {
this.username = username;
}
public String getPassword() {
return password;
}
public void setPassword(String password) {
this.password = password;
}
}</td><td>Only 60% Test cases are accepted</td></tr>
<tr><td>STATE MAP</td><td>import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Scanner;</td><td>public class Address {
private String addressLine1;</td><td></td></tr>
</table>

```
public class Main {

		public static void main(String[] args){
		Scanner sc = new Scanner(System.in);
	String Line =" ";

		List <Address> l = new ArrayList<>();
		Address ad;
		System.out.println("Enter the number of address");
		int n = sc.nextInt();
		sc.nextLine();
			for(int i=1;i<=n;i++) {

			System.out.println("Enter the address "+(i)+" detail");

			Line = sc.nextLine();
			String[] arr = Line.split(",");
			String addressL1 = arr[0];
			String addressL2 =arr[1] ;
			String City = arr[2];
			String State = arr[3];
		  int Pcode = Integer.parseInt(arr[4]);

		  ad = new Address(addressL1,addressL2,City,State,Pcode);
		  l.add(ad);

		}

		HashMap<String,List>  hs= new HashMap<>();
		Map<String,List>  hs2= new HashMap<>();
		Map<String,HashMap> mainhs = new HashMap<>();

		System.out.println("Enter the state to be searched");
		String searchstate = sc.nextLine();



		for (Address stu : l) {

	hs.put(stu.getState(), l);
	mainhs.put(stu.getCity(),hs);
    }

    if(hs.containsKey(searchstate)) {

	System.out.println("Enter the city to be searched");
		String searchCity = sc.nextLine();
	  if(mainhs.containsKey(searchCity)) {
			System.out.printf("%-15s %-15s %-15s %-15s %s\n","Line 1","Line
2","City","State","Pincode" );
			for (Address add : l) {
				if((searchstate.equals(add.getState()))&&(searchCity.equals(add.getCity())))
{
		System.out.printf("%-15s %-15s %-15s %-15s %s\n",
add.getAddressLine1(),add.getAddressLine2(),add.getCity(),add.getState(),add.getPincode());
				}
			}

	  }
	  else {
			System.out.println("Searched city not found");
	  }
    }
    else {
	System.out.println("Searched state not found");
    }


	  }

}
```

```
  private String addressLine2;
	private String city;
	private String state;
	private int pincode;



	public Address(String addressLine1, String addressLine2, String city, String state, int
pincode) {
		super();
		this.addressLine1 = addressLine1;
		this.addressLine2 = addressLine2;
		this.city = city;
		this.state = state;
		this.pincode = pincode;
	}
	public Address() {
		super();
		// TODO Auto-generated constructor stub
	}
	public String getAddressLine1() {
		return addressLine1;
	}
	public void setAddressLine1(String addressLine1) {
		this.addressLine1 = addressLine1;
	}
	public String getAddressLine2() {
		return addressLine2;
	}
	public void setAddressLine2(String addressLine2) {
		this.addressLine2 = addressLine2;
	}
	public String getCity() {
		return city;
	}
	public void setCity(String city) {
		this.city = city;
	}
	public String getState() {
		return state;
	}
	public void setState(String state) {
		this.state = state;
	}
	public int getPincode() {
		return pincode;
	}
	public void setPincode(int pincode) {
		this.pincode = pincode;
	}
}
```

GENERIC METHODS

```
import java.util.Scanner;
import java.io.*;

public class Main {
   public static < E > void printArray( E[] inputArray ) {
     // Display array elements
     for(E element : inputArray) {
       System.out.printf("%s ", element);
     }
     System.out.println();
   }
   public static void main(String[] args) throws IOException {



		Scanner sc = new Scanner(System.in);
		System.out.println("Enter a number :");
		int n = sc.nextInt();

		sc.nextLine();
		System.out.println("Enter the elements of the integer array");
		String m = sc.nextLine();
		String[] y = m.split(" ");
		Integer[] intArray = new Integer[n];

				for(int i=0; i<n; i++) {
			intArray[i] = Integer.parseInt(y[i]);
			}


		System.out.println("Enter the elements of the double array");
		String d = sc.nextLine();
		String[] z = d.split(" ");
		Double[] doubleArray = new Double[n];
```

```
                    for(int i=0; i<n; i++) {
                  doubleArray[i] = Double.parseDouble(z[i]);
                }

        System.out.println("Enter the elements of the character array");
        String x = sc.nextLine();

        String[] charArray= x.split(" ");


        System.out.println("Integer array contains:");
        printArray(intArray);

        System.out.println("Double array contains:");
        printArray(doubleArray);

        System.out.println("Character array contains:");
        printArray(charArray);
    }
}
```

Topic : Java Streams and Writers

I-Practice (Mandatory)

Question	Solution			Remarks

FILE HANDLING INTRODUCTION	import java.io.BufferedReader;
	import java.io.FileNotFoundException;
	import java.io.FileReader;
	import java.io.IOException;
	import java.util.List;
	
	public class Main {
	
	public static void main(String[] args){
	
	try {
	
	BufferedReader br=new BufferedReader(new FileReader("input.csv"));
	
	UserBO user = new UserBO();
	List<User> list =user.readFromFile(br);
	if(list.size()==0) {
	System.out.println("The list is empty");
	}else {
	System.out.printf("%-15s %-20s %-15s %s\n", "Name", "Email", "Username","Password");
	user.display(list);
	
	}
	}
	catch (FileNotFoundException e) {
	
	e.printStackTrace();
	
	} catch (IOException e) {
	
	e.printStackTrace();
	
	}
	
	}
	}
		public class User{
		
		private String name;
		private String email;
		private String username;
		private String password;
		public User(String name, String email, String username, String password) {
		super();
		this.name = name;
		this.email = email;
		this.username = username;
		this.password = password;
		}
		
		public String toString() {
		return String.format("%-15s %-20s %-15s %s\n",getName(),getEmail(),getUsername(),getPassword());
		}
		public User() {
		super();
		// TODO Auto-generated constructor stub
		}
		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 getUsername() {
		return username;
		}
		public void setUsername(String username) {
		this.username = username;
		}
		public String getPassword() {
		return password;
		}
		public void setPassword(String password) {
		this.password = password;
		}
		}
			import java.io.BufferedReader;
			import java.io.IOException;
			import java.util.ArrayList;
			import java.util.List;
			
			public class UserBO {
			
			public List<User> readFromFile(BufferedReader br) throws IOException{
			List<User> user = new ArrayList<User>();
			String line = null;
			
			while ((line = br.readLine()) != null) {
			String[] values = line.split(",");
			String name =values[0];
			String email =values[1];
			String username =values[2];
			String password =values[3];
			User u = new User(name, email, username, password);
			
			user.add(u);
			
			}
			return user;
			
			}
			
			public void display(List<User> list) {
			
			for(User t:list) // unboxing
			{
			
			System.out.printf("%-15s %-20s %-15s %s\n",t.getName(),t.getEmail(),t.getUsername(),t.getPassword());
			
			}
			}
			}	

FILE WRITING	import java.io.IOException;
	import java.util.List;
	import java.util.Scanner;
	import java.io.BufferedWriter;
	import java.io.File;
	import java.io.FileWriter;
	import java.util.ArrayList;
	import java.io.*;
	public class Main {
	public static void main(String args[]) {
	try{
	Scanner sc = new Scanner(System.in);
	ArrayList<User> users= new ArrayList<>();
	System.out.println("Enter the number of users:");
	int n = sc.nextInt();
	sc.nextLine();
	for(int i=0;i<n;i++) {
	System.out.println("Enter the details of user : "+(i+1));
	String input = sc.nextLine();
	String[] stringarray = input.split(",");
	String name = stringarray[0];
	String mobileNo = stringarray[1];
	String username =stringarray[3];
	String password=stringarray[3];
	users.add(new User(name,mobileNo,username,password));
		public class User {
		
		private String name;
		private String mobileNumber;
		private String username;
		private String password;
		public User(String name, String mobileNumber, String username, String password) {
		super();
		this.name = name;
		this.mobileNumber = mobileNumber;
		this.username = username;
		this.password = password;
		}
		public User() {
		super();
		// TODO Auto-generated constructor stub
		}
		public String getName() {
		return name;
		}
		public void setName(String name) {
		this.name = name;
		}
		public String getMobileNumber() {
		return mobileNumber;
		}
		public void setMobileNumber(String mobileNumber) {
			import java.io.BufferedWriter;
			
			import java.io.IOException;
			import java.util.ArrayList;
			import java.util.List;
			public class UserBO {
			
			public static void writeFile(ArrayList<User> userList, BufferedWriter bw) throws Exception {
			try {
			
			for(User t:userList) // unboxing
			{
			
			bw.write(t.getName()+","+t.getMobileNumber()+","+t.getUsername()+","+t.getPassword()+"\n");
			}
			
			bw.flush();
			bw.close();
			
			} catch (IOException e) {
			// TODO Auto-generated catch block
			e.printStackTrace();
			}
			}
			}	

	}
	
	File file = new File("output.csv");
	if (!file.exists()) {
	file.createNewFile();
	}
	
	FileWriter fw = new FileWriter(file);
	BufferedWriter bw = new BufferedWriter(fw);
	UserBO.writeFile(users, bw) ;
	
	} catch (IOException e) {
	// TODO Auto-generated catch block
	e.printStackTrace();
	}catch (Exception e) {
	// TODO Auto-generated catch block
	e.printStackTrace();
	}
	
	}
	}
		this.mobileNumber= mobileNumber;
		}
		public String getUsername() {
		return username;
		}
		public void setUsername(String username) {
		this.username = username;
		}
		public String getPassword() {
		return password;
		}
		public void setPassword(String password) {
		this.password = password;
		}
		}		

HALL DETAILS	import java.util.ArrayList;
	import java.util.List;
	import java.util.Scanner;
	
	public class Main {
	public static void main(String[] args){
	
	Scanner sc = new Scanner(System.in);
	List<Hall> hall = new ArrayList<>();
	System.out.println("Enter the number of halls:");
	int n = sc.nextInt();
	sc.nextLine();
	for(int i=0;i<n;i++) {
	
	String input = sc.nextLine();
	String[] stringarray = input.split(",");
	String name = stringarray[0];
	String contact = stringarray[1];
	double cost = Double.parseDouble(stringarray[2]);
	String owner =stringarray[3];
	hall.add(new Hall(name,contact,cost,owner));
	
	}
	
	Hall.writeHallDetails(hall);
	
	}
	
	}
		import java.io.BufferedWriter;
		import java.io.File;
		import java.io.FileWriter;
		import java.io.IOException;
		import java.util.List;
		
		public class Hall {
		private String name;
		private String contact;
		private Double costPerDay;
		private String owner;
		
		static void writeHallDetails(List<Hall> halls) {
		try {
		File file = new File("hall.csv");
		if (!file.exists()) {
		file.createNewFile();
		}
		
		FileWriter fw = new FileWriter(file);
		BufferedWriter bw = new BufferedWriter(fw);
		for(Hall t:halls) // unboxing
		{
		
		bw.write(t.getName()+","+t.getContact()+","+t.getCostPerDay()+","+t.getOwner()+"\n");
		}
		
		bw.flush();
		bw.close();
		
		} catch (IOException e) {
		// TODO Auto-generated catch block
		e.printStackTrace();
		}
		}
		
		public Hall(String name, String contact, Double costPerDay, String owner) {
		super();
		this.name = name;
		this.contact = contact;
		this.costPerDay = costPerDay;
		this.owner = owner;
		}
		public Hall() {
		super();
		// TODO Auto-generated constructor stub
		}
		public String getName() {
		return name;
		}
		public void setName(String name) {
		this.name = name;
		}
		public String getContact() {
		return contact;
		}
		public void setContact(String contact) {
		this.contact = contact;
		}
		public Double getCostPerDay() {
		return costPerDay;
		}
		public void setCostPerDay(Double costPerDay) {
		this.costPerDay = costPerDay;
		}
		public String getOwner() {
		return owner;
		}
		public void setOwner(String owner) {
		this.owner = owner;
		}
		
		}	

I-Practice (Bridges)				
Question	Solution			Remarks

READ AND WRITE

```
import java.io.*;
import java.util.*;
public class Main {

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

				BufferedReader br=new
BufferedReader(new FileReader("input.txt"));
				//store data to buffer
				EventBO event = new EventBO();
				List<Event> list
=event.readFromFile(br);
				if(list.size()==0) {
				   System.out.println("The list is
empty");
				}
	else {

	  File file = new File("output.txt");
	  if (!file.exists()) {
			file.createNewFile();
	    }

						FileWriter fr = new
FileWriter(file);
					event.writeFile(list, fr);
				}


	} catch (FileNotFoundException e) {
						// TODO Auto-
generated catch block

	e.printStackTrace();


	} catch (IOException e) {
					// TODO Auto-
generated catch block

	e.printStackTrace();


	}

	}
}
```

```
import java.util.Comparator;
public class Event implements Comparator{
   private String eventName;
   private  int attendeesCount;
   private String ownerName;


  public String getEventName() {
		return eventName;
		}
		public void setEventName(String
eventName) {
			this.eventName =
eventName;
		}
		public int getAttendeesCount() {
			return attendeesCount;
		}
		public void
setAttendeesCount(int attendeesCount) {
			this.attendeesCount =
attendeesCount;
		}
		public String getOwnerName() {
			return ownerName;
		}
		public void
setOwnerName(String ownerName) {
			this.ownerName =
ownerName;
		}
		public Event(String eventName,
int attendeesCount, String ownerName) {
			super();
			this.eventName =
eventName;
			this.attendeesCount =
attendeesCount;
			this.ownerName =
ownerName;
		}
		public Event() {
			super();
			// TODO Auto-generated
constructor stub
		}

   //@Override
		public int compare(Object o1,
Object o2) {

			Event s1=(Event)o1;
			Event s2=(Event)o2;
			String
st1=s1.getOwnerName();
			String
st2=s2.getOwnerName();
			return st1.compareTo(st2);

		}

}
```

```
import java.io.*;
import java.util.*;

public class EventBO {

   public List<Event> readFromFile(BufferedReader br) throws IOException{

		List<Event> event = new ArrayList<Event>();
		String line = null;

      while ((line = br.readLine()) != null) {
         String[] values = line.split(",");
         String eventname =values[0];
           int attendeesCount =Integer.parseInt(values[1]);
           String ownerName =values[2];

         Event e = new Event(eventname,attendeesCount,ownerName);

         event.add(e);

      }
		return event;
	}
	void writeFile(List<Event> eventList,FileWriter fr) {

      try {
		BufferedWriter bw = new BufferedWriter(fr);

		    for(Event t:eventList)   // unboxing
		    {

bw.write(t.getEventName()+","+t.getAttendeesCount()+","+t.getOwnerName()
+"\n");
		    }
		    bw.flush();
		    bw.close();

		} catch (IOException e) {

				e.printStackTrace();
			}


	}
}
```

Only 25% Test Cases Are Acccepted

ITEM COUNT

```
import java.util.ArrayList;
import java.util.*;
import java.util.Map.Entry;
import java.io.*;
public class Main {
	public static void main(String args[]) {

		ArrayList<ItemType> itemTypeList=new ArrayList<>();
		itemTypeList.add(new ItemType("Electronics",new
Double(1000),new Double(100)));
		itemTypeList.add(new ItemType("Furniture",new Double(1000),new
Double(100)));
		itemTypeList.add(new ItemType("Decorations",new
Double(1000),new Double(100)));

		try {
	BufferedReader br=new BufferedReader(new FileReader("input.csv"));
		//store data to buffer
		ItemBO item = new ItemBO();
		ArrayList<Item> list =item.readFile(br,itemTypeList);
		TreeMap<String,Integer> tree = new TreeMap<>();
		if(list.size()==0) {
		   System.out.println("The list is empty");
		}
	else {

			tree=item.listItem(list,itemTypeList);
			System.out.println("The item types along with count of
each:");
			for(Entry<String, Integer> entry:tree.entrySet()){
		  System.out.println(entry.getKey()+"--"+entry.getValue());
		}

		}
		} catch (FileNotFoundException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
		} catch (IOException e) {
				// TODO Auto-generated catch block
				e.printStackTrace();
		}



	}
```

```
public class Item {

   private String number;
   private ItemType itemType;
	private String vendor;



	public Item(String number, ItemType itemType, String vendor) {
		super();
		this.number = number;
		this.itemType = itemType;
		this.vendor = vendor;
	}
	public Item() {
		super();
		// TODO Auto-generated constructor stub
	}
	public String getNumber() {
		return number;
	}
	public void setNumber(String number) {
		this.number = number;
	}
	public ItemType getItemType() {
		return itemType;
	}
	public void setItemType(ItemType itemType) {
		this.itemType = itemType;
	}
	public String getVendor() {
		return vendor;
	}
	public void setVendor(String vendor) {
		this.vendor = vendor;
	}

}
```

}

```
public class ItemType {


   private String name;
   private Double deposite;
         private Double costPerDay;



         public ItemType(String name, Double deposite, Double costPerDay) {
                  super();
                  this.name = name;
                  this.deposite = deposite;
                  this.costPerDay = costPerDay;
         }
         public ItemType() {
                  super();
                  // TODO Auto-generated constructor stub
         }
         public String getName() {
                  return name;
         }
         public void setName(String name) {
                  this.name = name;
         }
         public Double getDeposite() {
                  return deposite;
         }
         public void setDeposite(Double deposite) {
                  this.deposite = deposite;
         }
         public Double getCostPerDay() {
                  return costPerDay;
         }
         public void setCostPerDay(Double costPerDay) {
                  this.costPerDay = costPerDay;
         }


}
```

```
import java.util.*;
import java.util.Map.Entry;
import java.io.*;
public class ItemBO {

         public static ArrayList<Item> readFile(BufferedReader br,ArrayList<ItemType> typeList) throws
IOException {

       ArrayList<Item> item = new ArrayList<Item>();
       String line = null;

       while ((line = br.readLine()) != null) {
          String[] values = line.split(",");
          String number =values[0];
            String name =values[1];
            String vendor =values[2];

      for(ItemType it:typeList){
         if(it.getName()==name){
            Item e = new Item(number,it,vendor);

             item.add(e);
         }
      }


       }
                        return item;

             }

         public static TreeMap<String,Integer> listItem(ArrayList<Item> itemList,ArrayList<ItemType>
itemTypeList) {

      int count=0;
           TreeMap<String,Integer> tm = new TreeMap<>();

               for (ItemType i:itemTypeList) {
           Integer j = tm.get(i.getName());
           tm.put(i.getName(), (j == null) ? 1 : j + 1);
       }

                        return tm;


         }

}
```

I-Practice (Hots)

Question	Solution		Remarks

ITEM DETAILS	import java.util.*;
	import java.io.*;
	
	public class Main {
	public static void main(String[] args) throws IOException {
	BufferedReader br = new BufferedReader(new FileReader("input.csv"));
	
	ItemBO ib = new ItemBO();
	List<Item> item = new ArrayList<>();
	ItemType itemty ;
	item =ib.readFile(br);
	System.out.format("%-20s %-20s %-20s %s\n", "Item Number","Vendor Name","Item Type","Cost");
	
	for(Item i:item) {
	itemty = i.getItemType();
	System.out.format("%-20s %-20s %-20s %s\n",i.getItemNumber(),i.getVendor(),itemty.getItemTypeName(),itemty.getCost());
	}
	}
	}
		import java.util.*;
		import java.io.*;
		public class ItemBO {
		
		List<Item> readFile(BufferedReader br) throws IOException{
		String Line = null;
		Item i;
		ItemType it;
		List<Item> al = new ArrayList<>();
		while((Line =br.readLine())!=null) {
		String [] arr = Line.split(",");
		Integer Number=Integer.parseInt(arr[0]);
		String vendor =arr[1];
		String Type = arr[2];
		Double cost = Double.parseDouble(arr[3]);
		it = new ItemType(Type,cost);
		i = new Item(Number,vendor,it);
		
		al.add(i);
		
		}
		return al;
		
		}
		}	

	public class Item {
	
	private Integer itemNumber;
	private String vendor;
	private ItemType itemType;
	
	public Item(Integer itemNumber, String vendor, ItemType itemType) {
	super();
	this.itemNumber = itemNumber;
	this.vendor = vendor;
	this.itemType = itemType;
	}
	public Item() {
	super();
	// TODO Auto-generated constructor stub
	}
	public Integer getItemNumber() {
	return itemNumber;
	}
	public void setItemNumber(Integer itemNumber) {
	this.itemNumber = itemNumber;
	}
	public String getVendor() {
	return vendor;
	}
	public void setVendor(String vendor) {
	this.vendor = vendor;
	}
		public class ItemType {
		private String itemTypeName;
		private Double cost;
		
		public ItemType(String itemTypeName, Double cost) {
		super();
		this.itemTypeName = itemTypeName;
		this.cost = cost;
		}
		public ItemType() {
		super();
		// TODO Auto-generated constructor stub
		}
		public String getItemTypeName() {
		return itemTypeName;
		}
		public void setItemTypeName(String itemTypeName) {
		this.itemTypeName = itemTypeName;
		}
		public Double getCost() {
		return cost;
		}
		public void setCost(Double cost) {
		this.cost = cost;
		}
		}	

	public ItemType getItemType() {
	return itemType;
	}
	public void setItemType(ItemType itemType) {
	this.itemType = itemType;
	}
	}		

IO - SIMPLE FILE WRITE	import java.util.*;
	import java.io.*;
	public class Main {
	
	public static void main(String[] args) {
	
	Scanner sc = new Scanner(System.in);
	FileWriter fWriter = null;
	BufferedWriter writer = null;
	
	System.out.println("Enter the name of the airport");
	String name = sc.nextLine();
	
	System.out.println("Enter the city name");
	String city = sc.nextLine();
	
	System.out.println("Enter the country code");
	String code = sc.nextLine();
	
	try {
	fWriter = new FileWriter("airport.csv");
	writer = new BufferedWriter(fWriter);
	writer = new BufferedWriter(fWriter);
	writer.write(name);
	writer.write(",");
	writer.write(city);
	writer.write(",");
	writer.write(code);
	writer.newLine();
	writer.close();
	
	} catch (Exception e) {
	e.printStackTrace();
	}
	
	}
	
	}		