SAKSHI KALLUR
3B - OOJ LAB
1BM22CS345

LAB - 1

Sample Program.

(1) Program for printing "Hello"

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

→ output : Hello

(2) Program for prime number

```
class Main {
    public static void main(String a[])
    {
        int num=5;
        int flag;
        if (num == 0 || num == 1)
        {
            system.out.println(num + "is not a prime number");
        }
        else
        {
            for(i=2; i<=num/2; i++)
            {
                if(num % i == 0)
                {
                    flag=0;
                    break;
                }
            }
        }
        if(flag==0)
            system.out.println(num + "is not a prime number");
        else
            system.out.println(num + "is a prime number");
    }
```

OUTPUT :- 3 is a prime number

2 is not a prime number

(3) Program for Fibonacci series.

→
```
class Main{
    public static void main(String a[])
    {
     int n=10, n1=0, n2=1; int n3;
     System.out.print("Fibonacci series till "+n+
                                    "terms");
    for(int i=3; i<=n; i++)
    {
      n3 = n1+n2;
      System.out.print("  "+n3);
      n1=n2;
      n2=n3;
    }
  }
}
```

OUTPUT:- 0 1 1 2 3 5 8 13 21 34

(4) Program for addition, subtraction, multiplication, division.

→
```
class Main{
    public static void main(String a[])
  {
    int a=10 ; b=2;
    System.out.println(a+"-"+b+"is"+a+b);
    System.out.println(a+"-"+b+"is"+a-b);
    System.out.println(a+"*"+b+"is"+a*b);
    if(b==0)
      System.out.println("divide by zero error");
    else
      System.out.println(a+"/"+b+"is"+a/b);
  }
}
```

OUTPUT:- 11+4 is 15
5-3 is 2
4*3 is 12
10/5 is 2

LAB-2:

Q) Program to demonstrate constructor overloading by taking user input.

→
```
import java.util.Scanner
class Grocery
{
  double dal, pulse, sugar;
Grocery()
{
  double dal = pulse = 1;
        sugar = 0.5;
}
Grocery (double a)
{
   dal= pulse = sugar =a;
}
Grocery (double a, double b, double c)
{
  dal = a;
  pulse = b;
  sugar = c;
}
Grocery (Grocery obj)
{
   dal = obj.dal
   pulse = obj.pulse
   sugar = obj.sugar
}
void amount()
{
 System.out.print( dal*10 + pulse*20 + sugar*30);
}
}
Class Main.
{
 Public static void main (String a[])
 {
  Scanner sc = new Scanner (System.in);
  System.out.print (Enter quantity);
  double q = sc.nextDouble();
  System.out.print (Enter 3 quantities);
  double q1 = sc.next Double();
  double q2 = sc.next Double();
```

```
        double q3 = sc.nextDouble();
        Grocery g1 = new Grocery();
        Grocery g2 = new Grocery();
        Grocery g3 = new Grocery(q1, q2, q3);
        Grocery g4 = new Grocery(g3);
        g1.amount();
        g2.amount();
        g3.amount();
        g4.amount();
    }
}
```

OUTPUT :-

```
1500
2450
1350
1800
```

(4

(a)

Quadratic Equation

```
import java.util.Scanner;
class Quad
{
    int a, b, c;
    double r1, r2, d;
    void input()
{
    Scanner sc = new Scanner(System.in)
    a = sc.nextint();
    b = sc.nextint();
    c = sc.nextint();
}
void compute()
{
    if (a==0 || b==0 || c==0)
{
    System.out.println("invalid input");}
    d = (b*b) - (4*a*c);
    if (d==0)
```

```
{
    r1 = (-b)/(2*a);
    System.out.println("roots are real and equal");
    System.out.println("root1 = root 2 = " + r1);
}
else if (d<0)
{
    System.out.println("roots are imagnary");
    r1 = (-b) / (2*a);
    r2 = (Math.sqrt(-d)) / (2*a);
    System.out.println("roots = " + r1 + " + i " + r2);
    System.out.println("roots = " + r1 + " - i " + r2);
}
else if (d>0)
    r1 = (((-b + (Math.sqrt(d))) / double) (2*a);
    r2 = (((-b) + (Math.sqrt(d))) / double) (2*a);
    System.out.println("roots are equal and
                                    distinct");
    System.out.println("root1 = " + r1 + "root 2 = " + r2);
}
}
class QuadRun
{
    Public static void main (String a[])
    {
        Quad q = new Quad();
        q.input();
        q.compute();
    }
}
```

OUTPUT :- Enter value of a = 1
Enter value of b = 1
Enter value of c = 1
Roots are not real.

Enter value of a = 1
Enter value of b = 9
enter value of c = 1
Roots are :- 0.1125 and -8.887
Roots are real and distinct.

3 [2] Student

```
import java.util.Scanner;
class student
{
    String USN;
    String name;
    int marks[] = new int [6];
void accept_Details()
{
Scanner sc = new Scanner(System.in);
System.out.println("Enter USN: ");
USN = sc.nextLine();
SOP("Enter name: ");
name = sc.nextLine();
SOP("Enter marks of 6 subjects.");
for (int i=0; i<6 ; i++)
    {
        marks[i] = sc.nextInt();
    }
void calculatePercentage()
    {
      int totalMarks = 0;
      for (int mark : marks)
        {
          total marks += marks
        }
         percentage = (totalMarks/600) * 100;
    }
class Main{
   public static void main(String[] args)
{
  Scanner sc = new Scanner(System.in)
  System.out.println("Enter number of students: ");
  int numStudents = sc.nextInt();
  student[] students = new student[numStudents];
for (int i=0; i<numStudents; i++)
  {
    Students[i] = new Student();
```

```
System.out.println("Enter student details "+(i+1)+":");
students[i].acceptDetails();
students[i].calculatePercentage();
}
System.out.println("Student Details");
for(Student student : students)
   {
     student.displayDetails();
   }
}
```

OUTPUT:-

```
Enter number of students: 2
Enter USN: 1BM22EE009
Enter name: ABCD
Enter marks for 6 subjects:
98
95
94
92
97
99
Enter USN = 1BM22AI001
Enter name = EFGHI
Enter marks for 6 subjects:
87
86
90
99
97
95
Student Details:
USN: 1BM22EE009
Name: ABCD
Marks: 98,95,94,92,97,99
Percentage: 95.8%

USN: 1BM22AI001
Name: EFGHI
Marks: 87,86,90,99,97,95
Percentage: 92.3%.
```

Q1] Program to create abstract class named shape that contains two integer and empty method printArea(). Provide three classes Rectangle, Triangle and circle such that each one classes extends the class shape. Each one of the classes contains method printArea() that prints the area of given shape.

→
```
abstract class shape {
protected int length;
protected int breadth;
public Shape (int length, int breadth)
{
    this.length = length;
    this.breadth = breadth;
}
public abstract void printArea();
}
class Rectangle extends Shape {
        public Rectangle (int length, int breadth)
         super(length, breadth);
    }
    public void printArea()
    {
      int area = length * breadth;
      System.out.println ("Rectangle Area: " + area);
    }
}
  class Triangle extends Shape{
        public Triangle (int length, int breadth){
          super (length, breadth);
      }
      public void printArea(){
          double area = 0.5* length * breadth;
          SOP ("Triangle Area": + area);
      }
}
class Circle extends Shape {
      public circle (int radius) {
          super(radius, 0);
```

```
    public void printArea(){
        double area= Math.PI * length * length;
        System.out.println("Circle Area:"+Area);
    }
}
public class Main {
    public static void main(String[] args){
    Rectangle rectangle = new Rectangle(5,10);
    Triangle triangle = new Triangle(4, 10);
    Circle circle = new circle(3);
    rectangle.printArea();
    triangle.printArea();
    Circle.printArea();
    }
}
```

OUTPUT :-

```
Rectangle Area : 50
Triangle Area : 20.0
Circle Area : 28.274333 38823
```

Q2] BANK ACCOUNT

```
import java.util.Scanner;
class Account
{
  String customerName;
  long accountNumber;
  String account type;
  double balance;
  public Account (String customerName, long account number,
   String accounttype, double balance)
  {
   this.customerName = customerName;
   this.accountNumber = accountNumber;
   this.accountType = accounttype;
   this.balance = balance;
  }
public void deposit (double amount)
{
```

```
        balance += amount;
        SOP("Deposit successful");
        SOP("Updated balance: "+balance);
    }
    public void displayBalance()
{
        SOP("Account Number: " + accountNumber);
        SOP("CustomerName: "+customerName);
        SOP("Account Type: "+ accountType);
        SOP("Balance: " + balance);
    }
}
class SavAcct extends Account
 {
    public SavAcct(String customerName, long accountNumber,
        double balance)
    {
      SOP(customerName, accountNumber, "Savings",
                balance);
    }
public void computeAndDepositInterest(double rate)
{
    double interest = balance * rate /100%
    balance += interest;
    SOP("Interest computed and deposited");
    SOP("Updated balance: "+balance);
 }
public void withdraw(double amount)
{
    if(amount <= balance)
    {
      balance -= amount;
      SOP("Withdrawal successful")
      SOP("Updated balance: "+balance);
    }
  else
    {
      SOP("Insufficient funds");
      SOP("Updated balance: "+balance);
    }
```

```
}
}
public class CurrAcct extends Account
{
  double minimumBalance;
  double serviceCharge;
  public CurrAcct (String customerName, long account
   Number, double balance, double MinimumBalance,
   double serviceCharge)
{
  super (customerName, accountNumber, "Current", balance);
  this.minimumBalance = minimum Balance;
  this.serviceCharge = serviceCharge;
}
  private void checkMinimumBalance()
{
    if amount <= balance)
      (balance < minimumBalance)
  {
    Balance -= serviceCharge;
    SOP("Minimum balance not maintained")
    SOP("Service charge imposed")
    SOP("Updated balance: " + balance);
  }
}
Public void withdraw (double amount)
{
    if(amount <= balance)
  {
    balance -= amount;
    SOP("withdrawal successful")
    SOP("Updated balance: " + balance);
    checkMinimum Balance();
  }
  else
  {
    SOP("Insufficient funds, withdrawal failed");
  }
}
}
```

```
public class Bank
{
    psvm(String[] args)
{
    Scanner SI = new Scanner (System.in);
    SOP ("Enter customer name for savings Account:");
    String SCN = SI.nextline();
    System.out.println("Enter account Number for
            savings account");
    long SAN = SI.nextLong();
    SOP("Enter initial Balance for savings Account:");
     double SIB = SI.nextDouble();
    SavAcct SA = new SavAcct (SCN, SAN, SIB);
    SOP("Enter customer name for current Account:");
    String CCN = SI.next();
    SOP ("Enter account number for current Account:");
    long CAN = SI.nextLong();
    SOP("Enter initial balance for current Account");
    double CIB = SI.nextDouble();
    SOP("Enter minimum balance for current Account");
    double SC = SI.nextDouble();
   CurrAct CA = new CurrAct (CCN+CAN, CIB, MB, SC);
   SOP ("Enter deposit amount for savings Account:");
   double SDA = SI.nextDouble();
    SA.deposit(SDA);
   SOP("Enter interest rate for savings Account:");
    double SIR = SI.nextDouble();
    SA.ComputeAndDepositInterest(SIR);
   SOP("Enter deposit amount for current Account:");
   double CDA = SI.nextDouble();
   CA.deposit(CDA);
SOP("Enter withdrawal amount for current Account:");
double CWA = SI.nextDouble();
CA.withdrawal(CWA);
```

```
SOP ("Final Balance: ");
SOP (" Savings Account : ");
SA . displayBalance();
SOP(" Current Account : ");
CA . displayBalance();
}
}
```

OUTPUT :-

Enter customer name for Savings Account: Rita
Enter account number for savings Account : 25,000
Enter initial balance for savings Account : 10000
Enter customer name for current Account : Reya
Enter account number for current Account : 24000
Enter initial balance for current Account: 1000
Enter minimum balance for current Account: 5500
Deposit successful
Updated balance : 15,000
Enter interest rate for saving Account : 4
Interest computed and deposited.
Updated balance : 15,750.0
Enter withdrawal amount for savings Account : 700
Withdrawal successful
Updated balance : 15, 700.0.

SS
29/11/2021

Packages :-

8) Create a package CIE which has two classes - student and Internals. The class student has members like USN, name, sem. The class internals derived from student has an array that stores the internal marks scored in five courses of the current semester of student. Create another package SEE which has the class External which is a derived class of student. This class has an array that stores the SEE marks scored in five courses of the current semester of student. Import the 2 packages in a file that declares the final C marks of n students in all five courses.

Student.java

```
Package CIE
public class Student
{
    public string USN, name;
    public int sem;
    public Student (String USN, String name, int sem)
    {
        this.usn = usn;
        this.name = name;
        this.sem = sem;
    }
}
```

Internals.java

```
package CIE
public class Internals extends Student
{ public int m[] = new int[5];
  public Internals(String USN, String name, int sem,
                                                int[] m
  {
    super(USN, name, sem);
    this.m = m;
  }
}
```

Externals.java

```
package SEE
import CIE.Student;
public class External extends Student
{
  public int SM[] = new int[5];
  public External(String USN, String name, int sem,
                  int[] SM)
  {
    super(USN, name, sem);
    this.sm = sm;
  }
}
```

Main class.java

```
import java.util.Scanner;
import CIE.Student;
import CIE.Internals;
import SEE.External;
public class main class
{ public static void main(String args[])
  { int n=0;
    Scanner in = new Scanner(System.in);
    System.out.println("Enter no of Students");
    int n = in.nextInt();
    Internals[] im = new Internal[n];
    External[] em = new External[n];
    Student[] stu = new student[n];
    for(int i=0; i<n; i++)
    { System.out.println("Enter details for Student"+
                        (i+1));
      System.out.println("Enter Name:");
      in.nextline();
      String name = in.nextline();
      System.out.println("Enter USN:");
      String USN = in.nextline();
      System.out.println("Enter sem:");
      int sem = in.nextInt();
      int[] internal.marks = new int[5];
      int[] external.marks = new int[5];
      System.out.println("Enter marks Details");
```

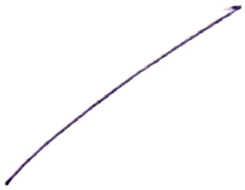
```
for (int j=20; j<t ;j +1)
{ SOP ("Enter internal marks for course " + (j+1));
  internal marks[j] = in.nextInt();
  SOP(" Enter external marks for course "+(j+1));
  external marks[j] = in. nextInt();
}
stu [i] = new student (USN, name, sem);
im [i] = new Internals (USN, name, sem, internal_mark
em[i] = new External (USN, name, sem, External_mar
}
SOP ("final marks Details :");
for (int i=0; i<n, i++)
{
  SOP ("student " + (i+1));
  SOP ("Name: " +stu[i]name);
  SOP ("USN" + stu[i].USN);
  SOP (" sem "+ stu[i]-sem);
  for (int j=0; j <5 ; j++)
  {
      tm = im[i]. m[j]+ em[i]. Sm[j];
      SOP (" final(marks" + (i+1) + e = ", tm);
      tm=0;
  }
}
}
```

OUTPUT:-

```
Enter no. of Students 1
Enter detail for student 1
Enter name: Bakshi
Enter USN : 035
Enter sem: 03
Enter marks details:
Enter internal marks for course 1 : 44
Enter Internal marks for Course 2: 38
Enter internal marks for course 2 : 41
Enter internal marks for course 2 : 42
```

Enter internal marks for course 3: 49
Enter internal marks for course 3: 41
Enter internal marks for course 4: 45
Enter internal marks for course 4: 48
Enter internal marks for course 5: 38
Enter internal marks for course 5: 43

Final Marks Details:
Student 1
Name: Sakshi
USN: 035
Sem: 03
Final Marks for course 1: 92
final marks for course 2: 85
final marks for course 3: 93
final marks for course 4: 87
Final marks for course 5: 92



19/02/2024

① Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In father class, implement a constructor which takes the age and throws WrongException() when the I/P age < 0. In class, implement a constructor that takes both father and son age and throws exception if son's age is >= father's age.

```
class WrongAge extends Exception
{
    public WrongAge (String message)
    {
        super (message);
    }
}
class Father
{
    int age;
    public father(int age) throws WrongAge
    {
        if (age < 0)
        {
            throw new WrongAge ("Age cannot be negative");
        }
        this.age = age;
    }
}
class son extends father
{
    int SonAge;
    public son(int fatherAge, int SonAge) throws WrongAge
    {
        super(fatherage);
        if (SonAge >= fatherAge) {
            throw new WrongAge Exception ("Son's age should be
                less than father's age");
        }
        this.SonAge = SonAge;
    }
    public int getSonAge() {
            return SonAge;
    }
}
```

```
public class ExceptionHandlingDemo {
    public static void main(String[] args) {
        try {
            Father father = new Father(40);
            Son son = new Son(father.getAge(), 20);
            System.out.println("Father's Age: " + father.getAge());
            System.out.println("Son's Age: " + son.getSonAge());
        }
        catch (WrongAgeException e){
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

OUTPUT :-

(1) Wrong age
WrongAge: Age cannot be negative
Wrong age
WrongAge: Son's age cannot be greater than or equal to father's age

(2) Write a program which creates two threads, one thread displaying "BMSCE" once every ten seconds and another displaying "CSE" once every two seconds.

```
public class BMS extends Thread
{
    public void run()
    {
        try
        {
            int i=0;
            while(i<100)
            {
                Thread.sleep(10000);
                System.out.println("BMSCE");
                i++;
            }
        }
    }
```

```
catch (InterruptedException e)
  {
    System.out.println("Interrupted");
  }
}
public void run()
{
   try
   {
    int i=0;
    while(i<100)
    {
       Thread.sleep(2000);
        System.out.println("CSE");
         i++;
    }
   }
  catch (Interrupted exception e)
   {
     System.out.println("Interrupted");
   }
}
public class MyThread
{  public static void main(String args[])
   {
     BMS b1 = new BMS();
     CS  c1 = new CS();
     b1.start();
     c1.start();
   }
}
```

OUTPUT:-

```
CSE
CSE
CSE
CSE
BMSCE
 CSE
CSE
CSE
```

CSE
CSE
BMSCE
CSE
BMSCE
BMSCE
BMSCE

DOD
19/2/24

1. Creating label button and Textfield in a frame using AWT

```
import java.awt.*;
import java.awt.event.*;
public class AWTExample extends WindowAdapter{
    frame f;
    AWTExample(){
        f = new frame();
        f.addWindowListener(this);
        Label l = new label("Employee id: ");
        Button b = new Button("Submit");
        Textfield t = new Textfield();
        l.setBounds(20,80,80,30);
        t.setBounds(20,100,80,30);
        b.setBounds(100,100,80,30);
        f.add(b);
        f.add(l);
        f.add(t);
        f.setsize(400,300);
        f.setTitle("Employee info");
        f.setLayout(null);
        f.setVisible(true);
    }
public void windowClosing(WindowEvent e){
    System.exit(0);
}
```

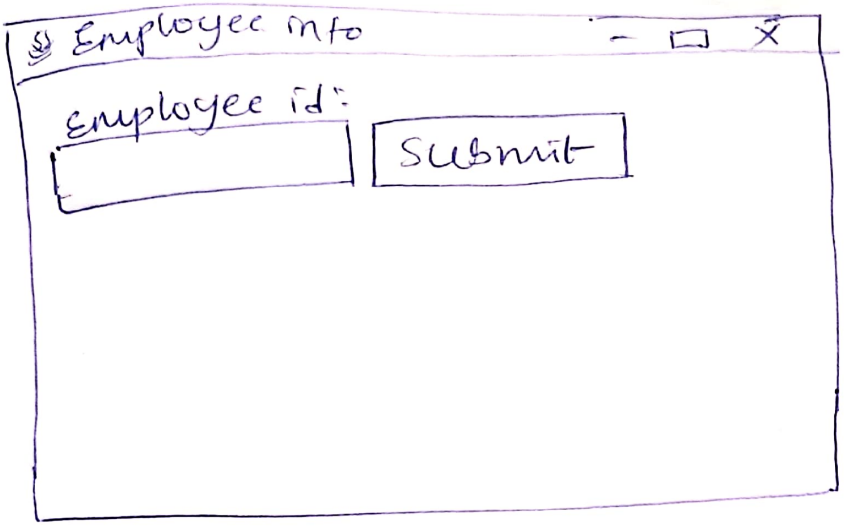
```
    public static void main (String[] args);
        AWTExample awt_obj = new AWTExample();
    }
}
```

2. Create a button and add an action listener for Mouse click.

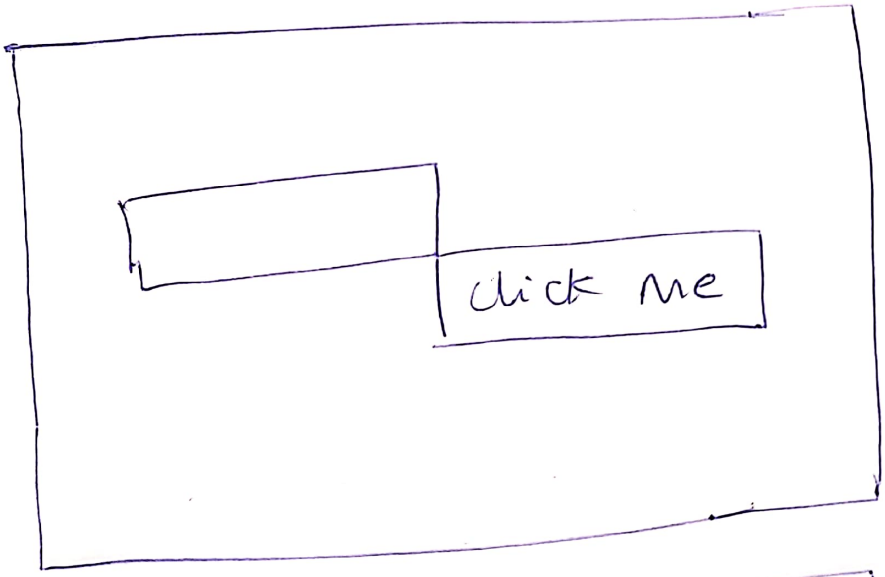
```
import java.awt.*;
import java.awt.event.*;
public class EventHandling extends WindowAdapter
    implements ActionListener{
        Frame f;
        TextField tf;
        EventHandling(){
        f. new Frame();
        f. addWindowListener (this);
        tf. new TextField();
        tf. addWindowListener
        tf. setBounds (60, 50, 170, 20);
        Button b = new Button ("Click me");
        b. setBounds (100, 120, 80, 30);
        b. addActionListener (this);
        f. add(b); f. add(tf);
        f. setSize(300, 300);
        f. setLayout (null);
        f. setVisible(true);
        }
        public void actionperformed (ActionEvent e){
         tf. setText("Welcome");
        }
        public void windowClosing (WindowEvent e);
                System. exit(0);
            }
        public static void main( String args[]);
            new EventHandling();
        }
}
```

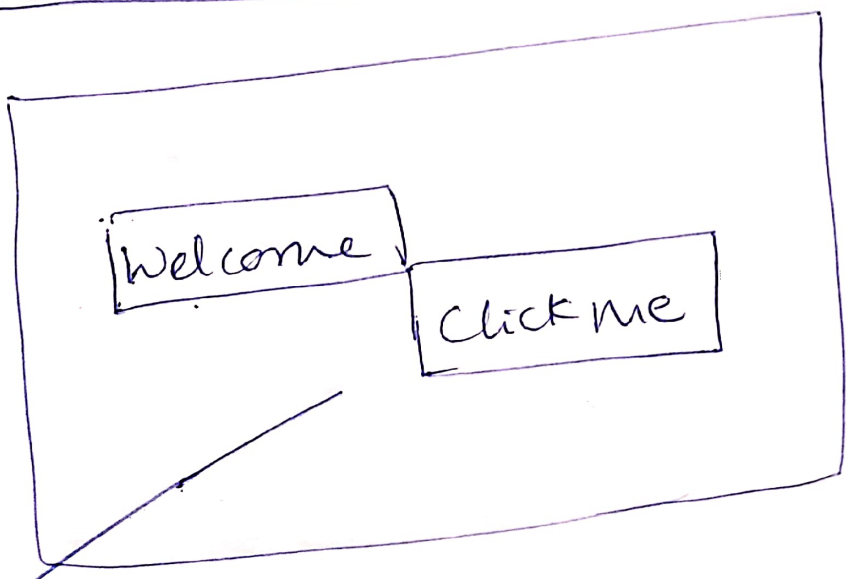
OUTPUT :-

1)



2)





*PROGRAMS ON IO

1. import java.io.*;
public class ByteArrayInput {
public static void main(String[] args) throws IOException {
byte[] buf = {35, 36, 37, 38};
ByteArrayInputStream byt = new ByteArrayInputStream(buf);
int k=0;
while((k = byt.read()) != -1) {
char ch = (char) k;
System.out.println("ASCII : "+k";
Special character: " +ch)}

2. public class FileEx{
public static void main (String a[]) throws IOException {
FileInputStream fin = new FileInputStream("ex.txt");
int content;
System.out.println("Remaining bytes that can be read: "+ fin.available());
content = fin.read();
System.out.print((char) content + " ");
System.out.print(content+" ");
System.out.println("Remaining bytes: "+fin.available());
System.out.println("Remaining bytes are: "+fin.available());
}
}

3. import java.io.FileInputStream;
import java.io.IOException;

public class FileEx2{
    public static void main(String a[]) throws
                IO Exception{
    FileInputstream fin=new FileInputstream(
            example.txt");
    byte[] bytes = new byte[20];
    int i;
    char c;
    i= fin.read(bytes);
    System.out.println("No of bytes read: "+i);
    System.out.print("Bytes read: ");
    for(byte b: bytes){
        c=(char) b;
        System.out.print(c);
    }
  }
}

3) BOOKS

```
import java.util.Scanner;
class Books
{
    String name;
    string author;
    int price;
    int numPages;
    Books(){}
    Books(string name, string author, int price, int
            numPages)
    {
      this.name = name;
      this.author = author;
      this.price = price;
```

```
    this.numPages = numPages; } public String toString()
  {
  String name, author, price, numPages;
  name = "Book name: " + this.name + "\n";
  author = "Book Author's name: " + this.author + "\n";
  price = "Price: " + this.price + "\n";
  numPages = "No. of pages: " + this.numPages + "\n";
  return name + author + price + numPages;
  }
}
class Main {
  public static void Main (String args[])
  {
   Scanner s = new Scanner (System.in);
   int n;
   String name;
   String author;
   int price;
   int numPages;
   System.out.println ("Enter the no of books: ");
   n = s.nextInt();
   Books b[] = new books[n];
   for (int i=0; i<n; i++)
   {
    System.out.println ("Book "+(i+1)+": ");
    System.out.println ("Enter name of book: ");
    name = s.next();
    System.out.println ("Enter author: ");
    author = s.next();
    System.out.println ("Enter price: ");
    price = s.nextInt();
   System.out.println ("Enter no of pages: ");
   numPages = s.nextInt();
   b[i] = new Books (name, author, price, numPages);
  }
    for(int i=0; i<n; i++)
        System.out.println ("Book"+(i+1)+":\n"+b[i]);
```

```
    }
}
```

OUTPUT:-

```
Enter the no of books: 2
Book 1:
Enter name of book: Jungle Book
Enter author: R Kipling
Enter price: 1000
Enter no of pages: 500

Book 2:
Enter name of book: Matilda
Enter author: Roald Dahl
Enter price: 900
Enter no of pages: 350

Book 1:
Book name: Jungle Book
Book's author name: R Kipling
price: 1000
no of pages: 500
Book 2:
Book name: Matilda
Book's author name: Roald Dahl
price: 900
no of pages: 350
```

SSD
26/2/24.