PAGE NO.:

* Print " hello world".

```
Public class Printmain {
    public static void main (String args[]) {
        System.out.println ("hello world");
    }
}
```

Output: 'hello world.'

2) Write a java program to multiply and divide two numbers

```
Public class Run {
    public static void main (String args[]) {
        int a = 6;
        int b = 2;
        System.out.println ("product is" + (a * b));
        if (b == 0) {
            System.out.println ("Not valid");
        }
        Else {
            System.out.println (a/b);
        }
    }
}
```

Output:

12
3

* Write a java program to add two numbers

```
Public class SumNum {
    public static void main (String args[]) {
        int a=5 ;
        int b=9 ;
        System.out.println (a+b);
    }
}
```

Output:

14.

Program 2

```
Public class Grocery {
    String c_name;
    string c_ph;
    double total;

    Grocery (String c_name, String c_ph) {
        this.c_name = c_name;
        this.c_ph = c_ph;
    }
    void Calc (double q_dal, double q_pulses,
        double q_sugar)
    {
        total = q_dal * 100 + q_pulses * 80 + q_sugar
            * 50;
    }
    void display ()
    {
        System.out.println ("Name" + " " + "phone number" +
            " " + "Total");
        System.out.println (c_name + " " + c_ph + " " + total);
        System.out.println ();
    }

    Class Gdemo {

    }
```

```
Public static void main (String [] args ) {
    Grocery g1 = new Grocery ("Rama", "8060302010");
    Grocery g2 = new Grocery ("shama", "76896325 10");
    Grocery g3 = new Grocery ("Bhama", "9632587412");
    g1. calc (2, 9, 1);
    g1. display ();
    g2. calc (3, 5, 9);
    g2. display ();
    g3. calc (1, 1, 0.5);
    g3. display ();
```

Output:

Name	Phone number	total
Rama	8060302010	410.0
Shama	76896325 10	800.0
Bhama	9632587412	205.0

- Program 1

Write a program to overload the print that prints sum of n natural numbers when one variable is passed and print the prime number in a given range. when 2 parameters are passed.

```
Public class Overload {
    void print (int n) {
        int sum = 0;
        for (int i=1 ; i <n ; i++) {
        sum = sum + i
        }
        System.out.println (" sum of "+n+" natural
        number is " + sum);
    }
    void print (int m, int n) {
    System.out.println (" prime number in the range
      are ");
    for (int i=m; i <=n ; i++) {
    int flag = 0;
    for(int j=2 ; j <= i/2 ; j++) {
    if (i%j == 0) {
    flag = 1
    break;
    }
    }
```

```
        if (flag ==0)
        System.out.println (i);
        }
        }
        }

        class overloadDemo {
    public static void main (String args[]) {
        Overload O = new overload ();
        O.print (5);
        O.print (7, 13);
    }
}
```

Output:

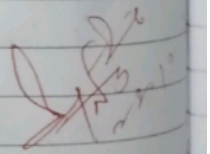
Sum of 5 natural number is 15
Prime numbers in the range are
7
11
13

PAGE NO.:

. Write a java program to calculate roots of a Quadratic equation. Use appropriate methods to take input, and calculate the roots.

```
import java.util.Scanner;

Public class Quad {
    int a, b, c;
    double root1, root2, d;
    Scanner s = new Scanner (System.in);
    void input()
    {
    System.out.println("Quadratic Equation is in the form
    : ax^2 + bx + c");
    Syetem.out.print("Enter a:")
    a = s.nextInt();
    System.out.print("Enter b:")
    b = s.nextInt();
    System.out.print("enter c:")
     c = s.nextInt();
    }
    void discriminant() {
    d = (b*b) - (4*a*c);
    }
    Void calculateRoots() {
    if (d > 0)
```

```
{
System.out.println("Roots are real & unequal");
root 1 = (-b + math.sqrt(d)) / (2*a);
root 2 = (-b - math.sqrt(d)) / (2*a);
System.out.println("First root is : " + root 1);
System.out.println("Second root is : " + root 2);
}
Else if (d == 0)
{
System.out.println("Roots are real & equal")
root 1 = (-b + math.sqrt(d)) / (2*a);
System.out.println("Root :" + root1);

Class Main {
public static void main (String args [])
Quad q = new Quad ();
q.input ();
q.discriminant ();
q.calculateRoots ();
}
}
```



Output:

```
Quadratic Equation is in the form : ax^2 + bx + c
Enter a : 1
Enter b : -2
Enter c : 1
```

Roots and Equal.

12/1/24

1) Create a class Book that contains four members: name, author, price and num_pages. include a constructor to set the values for the number. inclu method to set and get the details of the objec include a toString() method that could display the complete details of the book. Develop a java progra to create n books objects.

```
import java.util.Scanner;

Class Books {
    Scanner s = new Scanner(System.in);
    String name;
    String author;
    int price;
    int num pages;

    Book (string name, string author, int price, int
        numpages) {
        this.name = name;
        this.author = author;
        this.price = price;
        this.numPages = num Pages;
}
```

```
Public string to String () {
  String bookName = "Book Name:" + this.name + "\n";
  String authorName = "Author Name:" + this.author + "\n";
  String bookprice = "Price:" + this.price + "\n";
  String pages = "Number of Pages :" + this.numpages + "\n";
  return bookName + authorName + bookPrice + Pages;
}
}

Public class bookDemo {
  public static void main (String [] args) {
    Scanner s = new Scanner (System.in);
    int n ;
    System.out.println ("Enter the name of books:");
    n = s.nextInt ();
    books b[] = new books [n];
    for (int i = 0; i < n ; i++) {
    System.out.println ("\n details of Book." +(i+1));
    System.out.println(" Enter the name of Book:");
    String name = s.next ();
    System.out.println (" Enter the name of author:");
    String author = s.next ();
    System.out.println (" Enter the price:");
    int price = s.nextInt ();
    System.out.println(" Enter the number of pages:");
    int numpages = s.nextInt ();
```

```
        b[i] = new Books (name, author, price, numpages);
    }

    System.out.println("\n Details of all books:");
    for (books book : b)
    {
        System.out.println(book.toString());
    }
}
}
```

Output!

```
Enter the number of books :
2

details of Book_1 :
Enter the name of book :- Java
Enter the name of author :- abc
Enter the price :- 500
Enter the number of Pages :- 600
```

details of Book_2 :

Enter the name of Book :
OOJ
Enter the name of author
def
Enter the Price :
450
Enter the number of Pages :
600

Details of all books :
Book Name : Java
Author Name : abc
Price : 500
Number of pages : 600

Book Name : OOJ
Author Name : def
Price : 450
Number of Pages : 600

PAGE NO:

+ Write a java program to create a class student with members USN, name, marks (6 subjects). include methods to accept student details and marks, Also include a method to calculate the percentage and display appropriate details (Array of student object to be created).

```
import java.util.Scanner

Class student {
    String USN;
    String name;
    int [] marks = new int [6];

    Void acceptDetails () {
        Scanner scanner = new Scanner (System.in);

        System.out.println ("Enter USN:");
        USN = scanner.next ();

        System.out.println ("Enter name:");
        name = scanner.next ();

        Syston.out.println ("Enter marks of 6 subjects:");
```

```
for(int i = 0 ; i < 6; i++) {
System.out.println("Subjects "+(i+1)+": ");
marks[i] = scanner.nextInt();
}
}

double calculatePercentage() {
    int totalMarks = 0;

    for (int mark : marks) {
        totalMarks += mark;
    }
    return (double) totalMarks / 6;
}

void displayDetails() {
    System.out.println("Student details:");
    System.out.println("USN: " + USN);
    System.out.println("name: " + name);
    System.out.println("Percentage: " + calculatePercentage() + "%");
}
}

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

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

System.out.println("Enter the number of students:");
int numStudents = scanner.nextInt();

Student[] students = new Student[numStudents];

for (int i = 0; i < numStudents; i++) {
    students[i] = new Student();
    System.out.println("\nEnter details for student "
    + (i+1) + ":");
    students[i].acceptDetails();
}

    System.out.println("\nDetails of all students:");
    for (Student student : students)
    {
        student.displayDetails();
        System.out.println();
    }
}
}
```

Output:

Enter the number of students: 2

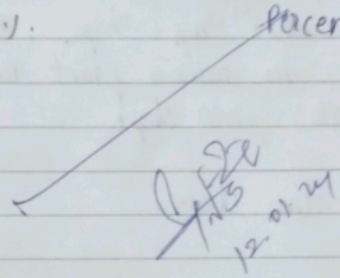
Enter details from student 1:
Enter USN: 2023BMS01
Enter name: Supreeth.
Enter marks for 6 subjects
80
90
90
90
90
90

Enter details for student 2:
Enter USN: 2023BMS02
Enter name: Faran
Enter marks for 6 subjects:
90
90
90
90
90
90

Details of all students :

Student details
USN : 2023BMS01
Name : Sajreeth.
Percentage = 88.3%.

Student details
USN : 2023BMS02
Name : Saran.
Percentage = 90.0%.



1. Develop a Java program to create an abstract named shape that contains two integer and a method named printArea(). Provide three classes Rectangle, Triangle and circle such that each extends the class shape. Each one of the class contain the method printArea. and print the area of the given shape.

```
abstract class shape.
{
  int x, y;
  abstract void area(double x, double y);
}

class Rectangle extends shape.
{
  void area (double x, double y);
  {
    System.out.println("Area of rectangle is :"+
  }
}
```

```
class Circle extends Shape
{
void area (double x, double y)
{
  System.out.println("Area of ~~triangle~~ circle is :" + (3.14 * x * x));
}
}

class Triangle extends Shape.
{
  void area (double x, double y)
  {
    System.out.println("Area of triangle is :" + (0.5 * x * y));
  }
}

Public class AbstractDemo {
    public static void main (String[] args) {
      Rectangle r = new Rectangle();
        r.area (3,5);
        Circle c = new Circle ();
        c.area (5,1);
        Triangle t = new Triangle ()
        t.area (2,5);
    }
}
```

Output:

Area of rectangle is : 10.0
Area of circle is : 78.5
Area of triangle is : 5.0

PAGE NO.:

2. Develop a java program to create class Bank that maintains two kind of account for its customer one called saving account and the other current account. the saving account provides compound interest and withdrawal facilities but no cheque book facility. the current account provides cheque book facility but no interest. Current account holder should also maintain a minimum balance and if the balance falls below the level, a service charge is imposed.

Create a class account that stores customer name account number and type of account from this derive the classes cur-acct and sav-acct to make them more specific to their requirements include the necessary methods in order to achieve the following tasks:

a. Accept deposit from customer and update the balance
b. Display the balance
c. Compute and deposit interest
d. Permit withdraw and update the balance.

check the minimum balance. impose penalty if necessary and update the balance.

```
class Account {
    protected String customerName;
    protected long accountNumber;
    protected String accountType;
    protected double balance;

    Public Account(String customerName, long accountNumber, String accountType, double balance) {
        this.customerName = customerName;
        this.accountNumber = accountNumber;
        this.accountType = accountType;
        this.balance = balance;
    }

    Public void displayBalance() {
        System.out.println("Account number: " + accountNumber);
        System.out.println("Customer Name: " + customerName);
        System.out.println("Account Type: " + accountType);
        System.out.println("Balance: $" + balance);
    }
    Public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposit of $" + amount + Succ
        displayBalance();
    }
```

```
Public void withdraw(double amount) {
if (amount <= balance) {
  balance -= amount;
  System.out.println("Withdrawal of $" + amount +
      "successfull.");
} display Else {
  System.out.println("Insufficient funds. Withdrawal
      failed.");
}

displayBalance();
}
}

Class Current extends Account {
  private double minimumBalance = 1000;
  private double serviceCharge = 50;

Public CurrAcct(String customerName, long accountNumber,
  double balance) {
  super(customerName, accountNumber, "Current",
      balance);
}
```

@ override

```
Public void withdraw (double amount) {
  if (amount <= balance - minimumBalance) {
     balance -= amount;
     System.out.println ("withdrawal of $" + amount +
       "successful.");
  }
  Else {
     System.out.println("Insufficient funds. withdrawal
     failed. Service charge of $" + serviceCharge + "applied.");
     balance -= serviceCharge;
  }
  displayBalance();
  }
}

class SavAcct extends Account {
  private double interestRate = 0.05;

  public SavAcct (String customerName, long accountNumber,
  double balance) {
     super (customerName, accountNumber, "saving",
     balance);
  }
```

}

```
public void ComputeInterest() {
    double interest = balance * interestRate;
    Balance += interest;
    System.out.println("Interest of $", interest +
        "credited");
    displayBalance();
}
}

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

    SAVAcct savingAccount = new SavAcct("John Doe",
        12345678, 500);
    SavingAccount.display();
    savingsAccount.deposit(1000);
    SavingsAccount.computeInterest();
    SavingsAccount.withdraw(2000);

    CurrAct currentaccount = new CurrAct("Jane Doe",
        98765432, 1500);
    currentAccount.display();
    currentAccount.deposit(500);
    CurrentAccount.withdraw(2000);

}}  scanner.close();
```

Output!

Account Number: 98765432.
Customer Name: Jane Doe
Account Type: Current.
Balance: $2000.0.
Insufficient funds. withdrawal failed. Service charge imposed.
Account Number: 98765432
Customer Name: Jane Doe.
Account Type: Current
Balance: $1950.0.

19.01.24

1) Create a package CIE which has two classes - Student has member 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 sem of the student. Create another package SEE which has class external which is a derived class of student. this class as an array that stores the SEE marks scored in five courses of the current semester of the student. Import two packages in a file that declares the final marks of n students in all five courses.

1. Create a folder CIE and save the program Student.Java & Internals.java within it.

2. Create a folder SEE and save the program External.java within it.

3. Save the main program outside these two folders.

4. Compile main.java and Execute the main class.

```
Package CIE ;

import java.util.Scanner;

Public class student {
protected String usn = new String ();
protected String name = new String ();
protected int sem ;

Public void inputStudentDetails () {
   Scanner s = new Scanner (System.in);
   System.out.println("give usn");
   usn = s.next Line ();
   System.out.println
```

---

1) CIE package

```
package CIE ;
public class Personal {
  public String usn;
  public String name;
  public int sem;
  public personal (String usn, String name, int sem)
  this.usn = usn;
  this.name = name
  this.sem = sem;
}
```

```
import java.util.arrays;
public class internals {
    public int[] internal marks;
    public internals (int[] internal marks) {
    this.internal marks = internal marks;
    }
}
```

```
Package SEE;
import CIE personal;
public class external extends personal {
    public int [] see marks;
    public External (String usn, String name, int sem,
    int [] see marks)
    super (usn, name, sem);
    this.see marks = see marks;
}
}
```

```
Package Formain;
import java.util.arrays;
import CIE.internals;
import CIE.personal;
import SEE.external;
```

```
public class main {
    public static void main (String [] args) {
     int n=3
     Student [] students = new student [n];
     for (int i=0 ; i<n ; i++) {
     int [] inter marks = {80, 75, 90, 85, 95};
     Int [] see marks = {30, 80, 75, 90, 85};
     Student [i]= new student (new personal ("usn"+i,
     "student" + i, 1) new internal (internal marks));
     student [i]. see = new external ("usn"+i, "student"
                    see marks);
     for (int i=0 ; i < students. length ; i++) {
       student student = student [i];
     SOP ("student :" + student . personal name);
     SOP ("internal marks" + Array to string ("student.
                internal , internal marks));
     SOP ("SEE marks" + Array to string (student.see
        marks ,
     SOP ();
    }
   }

Static class student {
public personal personal;
public internal internal;
public student (personal personal, Internal internal
```

this.personal = personal;
this.internals = internals;
}
}

Output:

student: Student 0
Internal marks: [80, 75, 90, 85, 95]
SEE marks: [70, 80, 75, 90, 85]

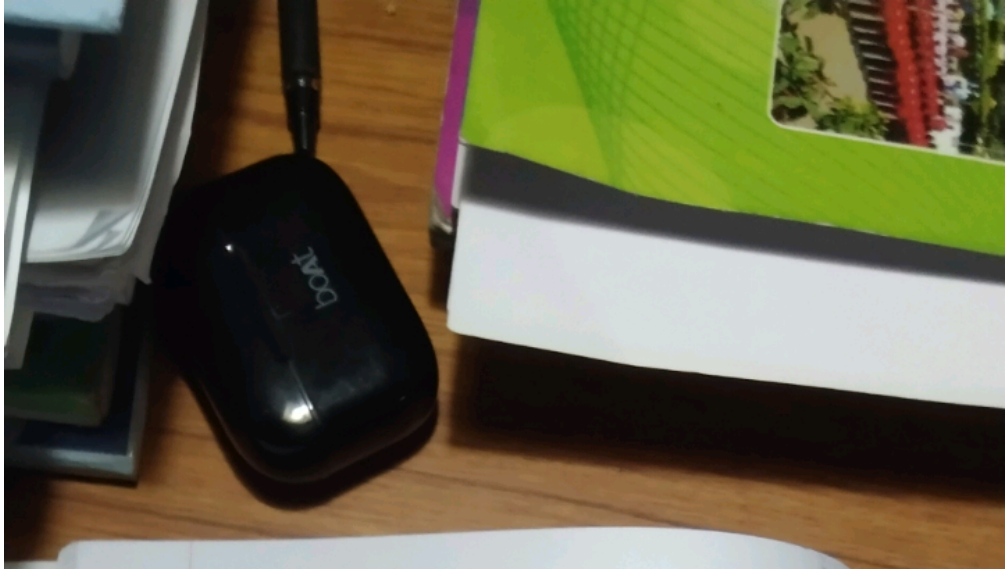
Student: Student 1
Internal marks: [80, 75, 90, 95, 90]
SEE marks: [70, 80, 75, 90, 85]

Student: Student 2
Internal marks: [80, 75, 90, 85, 95]
SEE marks: [70, 80, 15, 90, 85]

2) Write a program that demonstrates handling of exceptions in inheritence tree. Create 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 the exception wrong age(), when the input age < 0. In son class. Implement a constructor that cases both father and sons age and throws an exception if son's age is >= father's age.

```
class Father {
public int age;
father(int age) {
if (age > 0) {
  throw new illegal argument Exception("Age can
    be -ve");
};
this age = age;
}
}
public class son extends father {
  public int son age;
public son (int fatherage, int sonAge) {
super (father age)

if (son age >= father age) {
```

```
throw new IllegalArgumentException("son age cannot
    be >= father age");
}
this.sonage = sonage;
}
}

import java.util.Scanner;
public class Main {
  public static void main(String args[]){
  Scanner s = new Scanner(System.in);
  try {
    SOP("Enter father age:");
    int fatherage = s.nextInt();
    SOP("Enter son age:");
    int sonage = s.next.nt();
  Son son = new Son(fatherage, sonage);
  SOP("father age:" + fatherage);
  SOP("son's age:" + sonage);
}
catch (IllegalArgumentException e) {
  SOP("exception:" + e.getMessage());
}
  s.close();
}
}
```



output :

Enter father age : 20
enter son age : 40
Exception son's age cannot be >= father's age.

3)

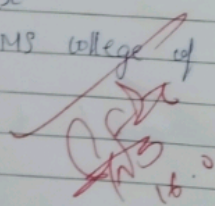
3) write a program which creates two threads, one thread displays "BMS college of Engineering" once every ten second and another displaying "CSE" once every two seconds.

```
Public class main {
public static void main (String [] args)
{
  thread thread 1 = new thread () {
    while (true) {
SOP ("BMS college of Engineering");
try {
    thread . sleep (10,000);
}
catch ( Interrupted exception e) {
      e. print stack trace ();
}
}
};

thread thread 2 = new thread () {
  while (true) {
SOP ("CSE");
try {
    thread. sleep (2000);
}
```

```
catch (InterruptedException e) {
        e.printStackTrace();
}
}

thread1.start();
thread2.start();
}
```

Output:

```
BMS college of Engineering
CSE
CSE
BMS college of Engineering
CSE
CSE
CSE
CSE
BMS college of Engineering
```



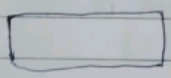
1. Create label button and text in a frame using AWT.

```
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 ();
}
}
```

Output :

Employee id :

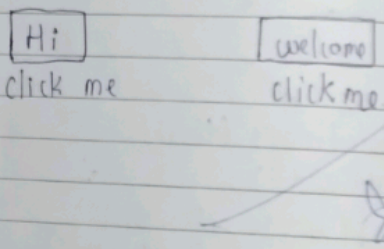


9. Create a button and add an action listener for mouse click

```
Import java.awt.*;
import java.awl.event.*;
public class EventHandling extends WindowAdapter
 implements action Listener {
    Frame f;
    TextField tf;
    EventHandling ()2

f = new Frame ();
f.addwindowListner (this);
Tf = new Textfield ();
Tf.setBounds (60,50,170,80);
Button b = new Button ("click me");
b.addActionListner (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"); system.exit (0)
}
Public static void main (String args [ ]) {
  new EventHandling ();
}
}
```

Output!



- Programs on IO

```
import java.io.*;
public class ByteArrayInput {
public static void main(String args[]) throws IO
Exception {
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 value of character is : "
+ k + "; special character is : " + ch);
}
}
}
```

Output:

```
Ascii of char 35  special char #
Ascii of char 36  special char $
Ascii of char 37  special char %
Ascii of char 38  special char &
```

8. import java.io.*;
public class ByteArrayInput {
public static void main (String args[]) throws
IOException {
byte[] buff = {35, 36, 37, 38};

ByteArrayInputStream byt = new ByteArrayInputStream
int k=0;
while ((k = byt.read()) != -1) {
char ch = (char) k;
System.out.println("ASCII value of character is "+
Special character is : " + ch);
}
}
}

3.

3. Public class FileEx {
public static void main (String a[]) throws
exception {
FileInputStream fin = new FileInputStream ("Example.txt");
int content;
System.out.println ("Remaining bytes that can be:"
+ fin.available());
System.out.println ("Remaining bytes than can be read
:" + fin.available());
}
}

```
import java.io.FileInputStream;
import java.io.IOException;

Public class FileEx2 {
  public static void main(String args[]) throws
  FileInputStream fin = new FileInputStream("Example
  byte []bytes = new byte[30];
  int i;
  char c;
  i = fin.read(bytes);
  SOP("Number of bytes read "+i);
  SOP("Bytes.read:");

  for(byte b:bytes) {
    c=(char)b;
    SOP(c);
  }
  }
  }
```