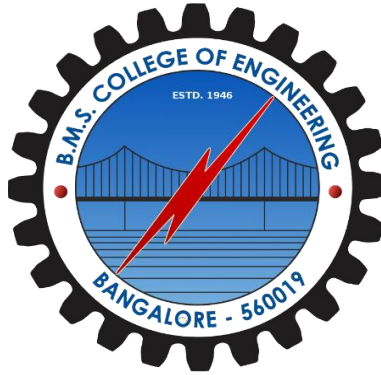


# **B.M.S. College of Engineering**

(Autonomous College Affiliated to Visvesvaraya Technological University,  
Belgaum) Bull Temple Road, Basavanagudi, Bengaluru – 560019



**Department of  
Computer Science & Engineering (CSE)**

## **Lab Programs Report**

**Course Title: Object Oriented**

**Java Programming**

**Course Code: 23CS3PCOOJ**

**BY**

**Tanush Prajwal S (1BM22CS304)**

# **B.M.S. College of Engineering**

(Autonomous College Affiliated to Visvesvaraya Technological University,  
Belgaum) Bull Temple Road, Basavanagudi, Bengaluru – 560019



**Department of  
Computer Science & Engineering (CSE)**

## ***CERTIFICATE***

This is to certify that the report on “**Java Lab Programs**” has been carried out by **Tanush Prajwal S** bearing USN **1BM22CS304** as a part of AAT for the course **Object Oriented Java Programming** with course code **23CS3PCOOJ**,

Computer Science and Engineering from Visvesvaraya Technological University, Belgaum during the year 2023–24. It is certified that all corrections/suggestions indicated for Internal Assessments have been incorporated in the report.

**Tanush Prajwal S**  
1BM22CS304

**Shravya AR**  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru-19

## Table of contents

<b>S. No.</b>	<b>Title</b>	<b>Pg No.</b>
1.	<b>Lab Program 1</b>	4-5
2.	<b>Lab Program 2</b>	6-7
3.	<b>Lab Program 3</b>	7-9
4.	<b>Lab Program 4</b>	9-11
5.	<b>Lab Program 5</b>	11-13
6.	<b>Lab Program 6</b>	14-16
7.	<b>Lab Program 7</b>	16-21
8.	<b>Lab Program 8</b>	21-23
9.	<b>Lab Program 9</b>	23-24

## LAB PROGRAM 1

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

### CODE

```
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 numbers is "+sum);
}

void print(int m, int n) {

System.out.println("Prime numbers 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(3,11);
}
```

### OUTPUT

Sum of 5 natural numbers is 15

Prime numbers in the range are

3

5

7

11

Process finished with exit code 0

## LAB PROGRAM 2

Write a Java program to create a class Grocery that has the variables c\_name and c\_phone. Create a method to accept 3 parameters to specify quantity of dal, quantity of pulses and quantity of sugar. The method to return the total price. Display the name, ph\_no and total bill of 3 customers

### CODE

```
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("Tanush","847583745");
        Grocery g2 = new Grocery("Tanush2","9454545360");
        Grocery g3 = new Grocery("Tanush3", "234598395");
        g1.calc(2, 2, 1);
        g1.display();
    }
}
```

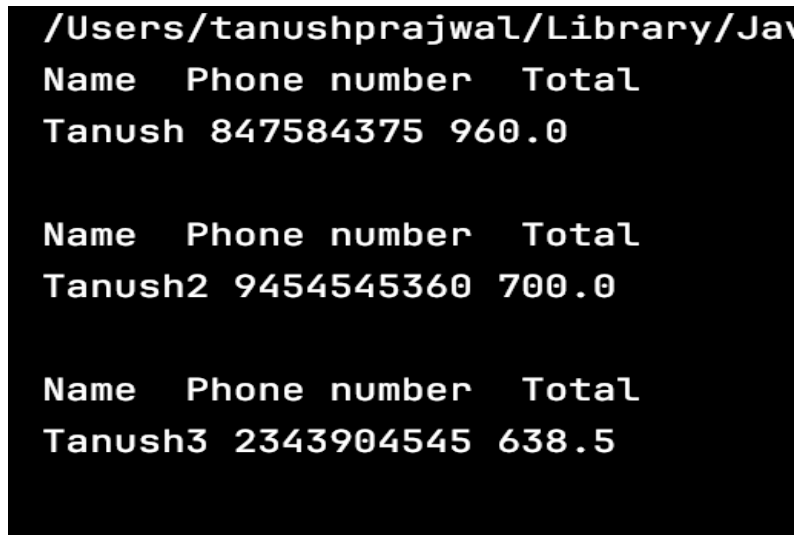
```
g2.calc(3, 5, 2);
g2.display();

g3.calc(1, 1, 0.5);
g3.display();

}

}
```

## OUTPUT



```
/Users/tanushprajwal/Library/Jav
Name  Phone number  Total
Tanush 847584375 960.0

Name  Phone number  Total
Tanush2 9454545360 700.0

Name  Phone number  Total
Tanush3 2343904545 638.5
```

## LAB PROGRAM 3

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

### CODE

```
import java.util.Scanner;

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");
```

```
System.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 and unequal");  
        root1 = ( - b + Math.sqrt(d))/(2*a);  
        root2 = (-b - Math.sqrt(d))/(2*a);  
        System.out.println("First root is:"+root1);  
        System.out.println("Second root is:"+root2);  
    }
```

```
    else if(d == 0)
```

```
    {  
        System.out.println("Roots are real and equal");
```

```
        root1 = (-b+Math.sqrt(d))/(2*a);  
        System.out.println("Root:"+root1);  
    }
```

```
    else
```

```
    {  
        System.out.println("No real solutions. Roots are imaginary");  
        double real = -b / (2 * a);  
        double imaginary = Math.sqrt(-d) / (2 * a);  
        System.out.println("The equation has two complex roots: " + real + " + " + imaginary + "i and "  
        + real + " - " + imaginary + "i");  
    }
```



```

}

}
class Main {

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

    }
}

```

## OUTPUT

```

/Users/tanushprajwal/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Contents/Home/bin/java
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:1
Enter c:1
No real solutions. Roots are imaginary
The equation has two complex roots: 0.0 + 0.8660254037844386i and 0.0 - 0.8660254037844386i

Process finished with exit code 0

```

```

/Users/tanushprajwal/Library/Java/JavaVirtualMachines/openjdk-21.0.1/Contents/Home/bin/java
Quadratic equation is in the form : ax^2 + bx + c
Enter a:1
Enter b:2
Enter c:1
Roots are real and equal
Root:-1.0

```

## LAB PROGRAM 4

Create a class Book that contains four members: name, author, price, and num\_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java program to create n book objects.

## CODE

```
import java.util.Scanner;

class Book{
    String name, author;
    int price, no_pages;

    public Book(String name, String author, int price, int no_pages){
        this.name = name;
        this.author = author;
        this.price = price;
        this.no_pages = no_pages;
    }

    public String toString(){
        System.out.println("Name: " + this.name);
        System.out.println("Author: " + this.author);
        System.out.println("Price: " + this.price);
        System.out.println("Pages: " + this.no_pages);
        return this.name + this.author + this.price + this.no_pages;
    }
}

class BookMain{
    public static void main(String args[]){
        System.out.println("Tanush Prajwal S - 1BM22CS304");
        Book books[] = new Book[10];
        Scanner sc = new Scanner(System.in);

        System.out.println("Enter no. of books: ");
        int n = sc.nextInt();
        sc.nextLine();
        for(int i = 0; i < n; i++){
            String name, author;
            int price, no_pages;
            System.out.println("Enter name: ");
            name = sc.next();
            System.out.println("Enter author: ");
            author = sc.next();
            System.out.println("Enter price: ");
            price = sc.nextInt();
            System.out.println("Enter no. of pages: ");
            no_pages = sc.nextInt();
            books[i] = new Book(name, author, price, no_pages);
        }
    }
}
```

```
}  
System.out.println("\n");  
for(int i = 0; i < n; i++){  
System.out.println("Book " + (i+1) + " Details:\n");  
books[i].toString();  
System.out.println("\n");  
}  
  
}
```

## OUTPUT

```
/Users/tanushprajwal/Library/Jav  
Tanush Prajwal S - 1BM22CS304  
Enter no. of books:  
1  
Enter name:  
abc  
Enter author:  
pqr  
Enter price:  
200  
Enter no. of pages:  
300  
  
Book 1 Details:  
  
Name: abc  
Author: pqr  
Price: 200  
Pages: 300
```

## LAB PROGRAM 5

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)

### CODE

```
import java.util.Scanner;

class Student {
    String USN;
    String name;
    int[] marks = new int[6];

    public void acceptDetails() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter USN:");
        USN = scanner.nextLine();
        System.out.println("Enter name:");
        name = scanner.nextLine();
        System.out.println("Enter marks for 6 subjects:");
        for (int i = 0; i < 6; i++) {
            System.out.print("Subject " + (i + 1) + ": ");
            marks[i] = scanner.nextInt();
        }
    }

    public double calculatePercentage() {
        int totalMarks = 0;
        for (int mark : marks) {
            totalMarks += mark;
        }
        return (double) totalMarks / 6;
    }

    public void displayDetails() {
        System.out.println("USN: " + USN);
        System.out.println("Name: " + name);
        System.out.println("Marks:");
    }
}
```

```

        for (int i = 0; i < 6; i++) {
            System.out.println("Subject " + (i + 1) + ": " + marks[i]);
        }
        System.out.println("Percentage: " + calculatePercentage() + "%");
    }
}

class Run {
    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++) {
            System.out.println("Enter details for student " + (i + 1) + ":");
            students[i] = new Student();
            students[i].acceptDetails();
        }

        System.out.println("\nDetails of students:");
        for (int i = 0; i < numStudents; i++) {
            System.out.println("\nStudent " + (i + 1) + ":");
            students[i].displayDetails();
        }
    }
}

```

## OUTPUT

Enter the number of students:

1

Enter details for student 1:

Enter USN:

304

Enter name:

abs

Enter marks for 6 subjects:

Subject 1: 67

Subject 2: 99

Subject 3: 98

Subject 4: 87

Subject 5: 77

Subject 6: 100

Details of students:

Student 1:

USN: 304

Name: abs

Marks:

Subject 1: 67

Subject 2: 99

Subject 3: 98

Subject 4: 87

Subject 5: 77

Subject 6: 100

Percentage: 88.0%

## LAB PROGRAM 6

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

### CODE

```
import java.util.Scanner;

class InputScanner{
    int d1, d2;
    Scanner sc = new Scanner(System.in);
    InputScanner(){
        if(this.getClass() == Circle.class){
            System.out.println("Enter d1: ");
            d1 = sc.nextInt();
        }
        else{
            System.out.println("Enter d1 and d2: ");
            d1 = sc.nextInt();
            d2 = sc.nextInt();
        }
    }
}

abstract class Shape extends InputScanner{
    abstract void printArea();
}

class Triangle extends Shape{
    void printArea(){
        System.out.println("Area of triangle is: " + (double)(d1*d2)/2);
    }
}

class Rectangle extends Shape{
    void printArea(){
        System.out.println("Area of rectangle is: " + (double)(d1*d2));
    }
}

class Circle extends Shape{
    void printArea(){
        System.out.println("Area of circle: " + (double)(3.14*d1*d1));
    }
}

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

```

Rectangle r = new Rectangle();
Triangle tr = new Triangle();
Circle c = new Circle();
r.printArea();
tr.printArea();
c.printArea();
}
}

```

### OUTPUT

```

Enter d1 and d2:
12
24
Enter d1 and d2:
24
12
Enter d1:
3
Area of rectangle is: 288.0
Area of triangle is: 144.0
Area of circle: 28.259999999999998

```

## LAB PROGRAM 7

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 the 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 the student. Import the two packages in a file that declares the final marks of n students in all five courses.

### CODE

```

package CIE;

public class Student {
    String usn;
    String name;
    int sem;

    public Student(String usn, String name, int sem) {
        this.usn = usn;
    }
}

```



```
        this.name = name;
        this.sem = sem;
    }
}
```

```
package CIE;
```

```
public class Internals extends Student {
    int[] internalMarks = new int[5];

    public Internals(String usn, String name, int sem) {
        super(usn, name, sem);
    }
}
```

```
package SEE;
```

```
import CIE.Student;
```

```
public class External extends Student {
    int[] externalMarks = new int[5];

    public External(String usn, String name, int sem) {
        super(usn, name, sem);
    }
}
```

```
public class CalculateFinalMarks {
    public static void main(String[] args) {
        Internals internalsStudent = new Internals("1BM22CS304", "Tanush", 3);
        internalsStudent.internalMarks = new int[]{80, 75, 85, 90, 95};

        External externalStudent = new External("1BM22CS444", "Random", 3);
        externalStudent.externalMarks = new int[]{70, 65, 75, 80, 85};

        for (int i = 0; i < 5; i++) {
            int finalMarks = internalsStudent.internalMarks[i] + externalStudent.externalMarks[i];
            System.out.println("Course " + (i + 1) + ": " + finalMarks);
        }
    }
}
```

## LAB PROGRAM 8

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this 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 withdrawal and update the balance Check for the minimum balance, impose penalty if necessary and update the balance.

### CODE

```
import java.util.*;
class Account{
    String name;
    int accno;
    String acc_type;
    double balance;

    Account(String name,int accno,String acc_type,double balance)
    {
        this.name=name;
        this.accno=accno;
        this.acc_type=acc_type;
        this.balance=balance;
    }

    void deposit(double amount)
    {
        balance+=amount;
    }

    void withdraw(double amount)
    {
        if((balance-amount)>=0)
        {
            balance-=amount;
        }
        else
        {
            System.out.println("Insufficient balance! Cannot withdraw");
        }
    }
}
```

```
}
```

```
void display()
```

```
{
```

```
System.out.println("Name:"+" "+name+"Account    number:"+" "+accno+"Account    type:"+" "+acc_type+"Balance:"+" "+balance);
```

```
}
```

```
}
```

```
class Sav_acct extends Account
```

```
{
```

```
static double rate=5.0;
```

```
Sav_acct(String name,int accno,double balance)
```

```
{
```

```
super(name,accno,"Savings",balance);
```

```
}
```

```
void interest()
```

```
{
```

```
balance+=balance*(rate)/100;
```

```
System.out.println("Balance:"+balance);
```

```
}
```

```
}
```

```
class Curr_account extends Account
```

```
{
```

```
private double minBal=500;
```

```
private double serviceCharges=50;
```

```
Curr_account(String name,int accno,double balance)
```

```
{
```

```
super(name,accno,"Current",balance);
```

```
}
```

```
void checkMin()
```

```
{
```

```
if (balance<minBal)
```

```
{
```

```
System.out.println("Balance is less than minimum, penlaty is imposed"+serviceCharges);
```

```
balance-=serviceCharges;
```

```
System.out.println("Balance is:"+balance);
```

```
}
```

```
}
```

```
}
```

```

class Bank
{
public static void main(String[] args)
{
Scanner s=new Scanner(System.in);
System.out.println("Enter customer name");
String name=s.next();
System.out.println("Enter account number");
int accno=s.nextInt();
System.out.println("Enter the type of account-Current or Savings");
String acc_type=s.next();
System.out.println("Enter the initial balance");
double balance=s.nextDouble();
Account ob1=new Account(name,accno,acc_type,balance);
Sav_acct sa=new Sav_acct(name,accno,balance);
Curr_account cu=new Curr_account(name,accno,balance);
while (true)
{
if(acc_type.equals("Savings"))
{
System.out.println("Menu");
System.out.println("1.Deposit");
System.out.println("2.Withdraw");
System.out.println("3.Compute Interest for savings account");
System.out.println("4.Display account details");
System.out.println("5.Exit");
System.out.println("Enter your choice:");
int choice=s.nextInt();

switch(choice)
{
case 1:
{
System.out.println("Enter the amount to be deposited");
double amt1=s.nextDouble();
sa.deposit(amt1);
break;
}
case 2:
{
System.out.println("Enter the amount to be withdrawn");
double amt2=s.nextDouble();
sa.withdraw(amt2);
break;
}
}
}
}
}

```

```

    }
    case 3:
    {
        sa.interest();
        break;
    }
    case 4:
    {
        sa.display();
        break;
    }
    case 5:
    {
        break;
    }
    default:
    {
        System.out.println("Enter valid input");
        break;
    }
}

}
else
{
    System.out.println("Menu");
    System.out.println("1.Deposit");
    System.out.println("2.Withdraw");
    System.out.println("3.Display account details");
    System.out.println("4.Exit");
    System.out.println("Enter your choice:");
    int choice=s.nextInt();
    switch(choice)
    {
    case 1:
    {
        System.out.println("Enter the amount to be deposited");
        double amt1=s.nextDouble();
        cu.deposit(amt1);
        break;
    }
    case 2:
    {
        System.out.println("Enter the amount to be withdrawn");

```

```
double amt2=s.nextDouble();
cu.withdraw(amt2);
break;
}
case 3:
{
cu.display();
cu.checkMin();
break;
}
case 4:
{
break;
}
default:
{
System.out.println("Enter valid input");
break;
}
}
}
}
}
```

## OUTPUT

```

/Users/tanushprajwal/Library/Java/JavaVi
Enter customer name
random
Enter account number
240034
Enter the type of account-Current or Sav
Savings
Enter the initial balance
100000
Menu
1.Deposit
2.Withdraw
3.Compute Interest for savings account
4.Display account details
5.Exit
Enter your choice:
1
Enter the amount to be deposited
30000
Menu
1.Deposit
2.Withdraw
3.Compute Interest for savings account
4.Display account details
5.Exit
Enter your choice:
3
Balance:136500.0
Menu
1.Deposit
2.Withdraw
3.Compute Interest for savings account
4.Display account details
5.Exit
Enter your choice:
3
Balance:143325.0
Menu
1.Deposit
2.Withdraw
3.Compute Interest for savings account
4.Display account details
5.Exit
Enter your choice:
4
Name: randomAccount number: 240034Account type: SavingsBalance: 143325.0
Menu
1.Deposit
2.Withdraw
3.Compute Interest for savings account
4.Display account details
5.Exit
Enter your choice:
5
Menu
1.Deposit
2.Withdraw

```

## LAB PROGRAM 8

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 the exception WrongAge( ) when the input age<0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is >=father's age.

## CODE

```
import java.util.Scanner;

class WrongAge extends RuntimeException {
    public WrongAge() {
        super("Age cannot be negative");
    }

    public WrongAge(String message) {
        super(message);
    }
}

class InputScanner {
    protected Scanner scanner;

    public InputScanner() {
        scanner = new Scanner(System.in);
    }

    public int nextInt() {
        return scanner.nextInt();
    }
}

class Father extends InputScanner {
    protected int fatherAge;

    public Father() {
        System.out.println("Enter father's age:");
        fatherAge = super.nextInt();

        if (fatherAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }

    public void display() {
        System.out.println("Father's Age: " + fatherAge);
    }
}

class Son extends Father {
    private int sonAge;

    public Son() {
        super();
        System.out.println("Enter son's age:");
        sonAge = super.nextInt();

        if (sonAge > fatherAge) {
            throw new WrongAge("Son's age cannot be greater than father's age!!!");
        } else if (sonAge < 0) {
            throw new WrongAge("Age cannot be negative");
        }
    }
}
```



```

public void display() {
    super.display();
    System.out.println("Son's Age: " + sonAge);
}
}

public class InheritanceException {
    public static void main(String[] args) {
        try {
            Son son = new Son();
            son.display();
        } catch (WrongAge e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}

```

## OUTPUT

```

/Users/tanushprajwal/Library/Java/JavaVirtualMachines/openjdk-?
Enter father's age:
40
Enter son's age:
55
Exception: Son's age cannot be greater than father's age!!!

Process finished with exit code 0

```

## LAB PROGRAM 9

Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds.

## CODE

```

import java.io.*;

class B extends Thread{
    public void run(){
        try{
            for(int i = 0; i < 3; i++){

```

```

System.out.println("BMS College of Engineering");
Thread.sleep(10000);
}
} catch (InterruptedException e){
System.out.println(e);
}

}
}

class C extends Thread{
public void run(){
try{
for(int i = 0; i < 3; i++){
System.out.println("CSE ");
Thread.sleep(2000);
}
} catch (InterruptedException e){
System.out.println(e);
}

}
}

class ThreadMain{
public static void main(String args[]){
B b = new B();
C c = new C();
b.start();
c.start();
}
}

```

## OUTPUT

/Users/tanushprajwal/Library/Java/Java

BMS College of Engineering

CSE

CSE

CSE

BMS College of Engineering

BMS College of Engineering

Process finished with exit code 0

**THANK YOU**