**Object oriented concepts Laboratory E-journal**

Goa Vidhyaprasarak Mandal’s

**GOPAL GOVIND POY RAITURCAR COLLEGE OF COMMERCE & ECONOMICS**



**FARMAGUDI PONDA GOA**

**BACHELOR OF COMPUTER APPLICATION**

CERTIFICATE

This is to certify that Shri. Chetan Badiger bearing Admission No: - 2305002 has successfully completed his work in Object oriented concepts Laboratory for his

semester 4 practical examination for the academic year 2024-2025

Internal examiner: -

External examiner: -

INDEX

|  |  |  |  |
| --- | --- | --- | --- |
| Sr.no | Date | Title | Page. No |
| 1 | 24/1/25 | basics of Java | 3 |
| 2 | 31/1/25 | Basic programs in Java | 6 |
| 3 | 7/2/25 | Factorial and Fibonacci programs | 10 |
| 4 | 14/2/25 | Programs using classes and objects | 14 |
| 5 | 21/2/25 | Constructors in Java | 17 |
| 6 | 28/2/25 | Intermediate programs in Java | 22 |
| 7 | 28/3/25 | inheritance and polymorphism | 30 |
| 8 | 4/4/25 | Exception handling | 35 |

Note: highly suggested to use light mode.

Title: basics of Java

Theory: Java's basic syntax is structured and easy to read, making it a good starting point for beginners. A simple Java program starts with a class definition, as everything in Java must be inside a class. The entry point of any Java application is the main method: public static void main(String[] args). Statements in Java end with a semicolon (;), and blocks of code are enclosed in curly braces ({} and }). Java is case-sensitive, meaning Main and main would be considered different. Variables must be declared with a type (e.g., int, String, double), and every method must be inside a class. Comments can be added using // for single-line or /\* \*/ for multi-line. Java follows a strict naming convention and encourages clarity and organization in code structure

# INPUT

package lab24\_1\_25;

public class helloworld {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

System.***out***.println("Hello World!");

}

}

# OUTPUT

Hello World!

# INPUT

package lab24\_1\_25;

public class plus\_operator {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int a=5,b=10;

int sum = a+b;

System.***out***.println(a+" + "+ b +" = "+ sum);

}

}

# OUTPUT

5 + 10 = 15

# INPUT

package lab24\_1\_25;

public class ari\_operators {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int a=5,b=10;

System.***out***.println(a + " + "+ b +" = "+(a+b));

System.***out***.println(a + " - "+ b +" = "+(a-b));

System.***out***.println(a + " \* "+ b +" = "+(a\*b));

System.***out***.println(a + " / "+ b +" = "+(a/b));

System.***out***.println(a + " % "+ b +" = "+(a%b));

}

}

# OUTPUT

5 + 10 = 15

5 - 10 = -5

5 \* 10 = 50

5 / 10 = 0

5 % 10 = 5

# INPUT

package lab24\_1\_25;

public class xyz\_operators {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int a=10,b=23,c=9 ;

System.***out***.println("X= " +(a\*b-c));

System.***out***.println("Y= " +(b/c\*a));

System.***out***.println("Z= " +(a-b/c+b));

}

}

# OUTPUT

X= 221

Y= 20

Z= 31

# INPUT

package lab24\_1\_25;

public class typecast {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int x= 2;

float y=4;

int sum=(int) y/x;

System.***out***.println("4.0/2 = "+(sum));

}

}

# OUTPUT

X= 221

Y= 20

Z= 31

Title: Basic programs in Java

Theory: Basic programs in Java include simple tasks. These programs help beginners understand core concepts such as syntax, data types, variables, and control structures.

# INPUT

package lab31\_1\_25;

public class Break {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int a=6;

for(int i=0;i<10;)

{

a+=1;

break;

}

System.***out***.println(a); //answer will be 7

}

}

# OUTPUT

7

# INPUT

package lab31\_1\_25;

public class Contine {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int a=6;

for(int i=0;i<10; i++)

{

a+=1;

continue;

}

for(int j=0;j<5; j++)

{

a+=1;

}

System.***out***.println(a); //answer will be 21

}

}

# OUTPUT

21

# INPUT

package lab31\_1\_25;

import java.util.Scanner;

public class Evenodd {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int a;

Scanner s = new Scanner(System.***in***);

a = s.nextInt();

if(a%2==0)

{

System.***out***.println("Even number");

}

else

{

System.***out***.println("Odd number");

}

s.close();

}

}

# OUTPUT

5

Odd number

# INPUT

package lab31\_1\_25;

public class Star\_pattern {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int rows=5;

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

{

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

{

System.***out***.print("\* ");

}

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

}

}

}

# OUTPUT

\*

\* \*

\* \* \*

\* \* \* \*

# INPUT

package lab31\_1\_25;

public class Volume {

public int volume(int width, int height, int depth) {

return width \* height \* depth;

}

public static void main(String[] args) {

Volume a = new Volume();

int res = a.volume(10, 8, 16);

System.***out***.println("The volume is: " + res);

}

}

# OUTPUT

The volume is: 1280

Title: Factorial and Fibonacci programs

Theory: Factorial and Fibonacci programs are common beginner exercises in Java that help understand loops and recursion. The factorial program calculates the product of all positive integers up to a given number (e.g., 5! = 5×4×3×2×1). It can be implemented using a loop or recursion. The Fibonacci program generates a series where each number is the sum of the two preceding ones (e.g., 0, 1, 1, 2, 3, 5…). These programs are great for learning function calls, logic building, and control flow in Java.

# INPUT

package lab07\_2\_25;

import java.util.Scanner;

class fact{

int ff (int n) {

int fact =1;

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

fact\*=i;

}

return fact;

}

}

public class Factorial {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

fact cal = new fact();

Scanner s =new Scanner(System.***in***);

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

int n = s.nextInt();

System.***out***.println(cal.ff(n));

s.close();

}

}

# OUTPUT

Enter a number

4

24

# INPUT

package lab07\_2\_25;

public class Fibonacci {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

int fib1 = 0;

int fib2 = 1;

int next;

for(int i=0;i<=10;i++)

{

next = fib1+fib2;

System.***out***.println(next);

fib1 = fib2;

fib2 = next;

}

}

}

# OUTPUT

1

2

3

5

8

13

21

34

55

89

144

# INPUT

package lab07\_2\_25;

import java.util.Scanner;

class read\_calculate\_display {

public double r() {

Scanner s = new Scanner(System.***in***);

System.***out***.println("Enter a length in cm:");

double len = s.nextInt();

System.***out***.println("Enter a breath in cm:");

double bre = s.nextInt();

s.close();

return len \* bre;

}

public void c() {

double res = r();

System.***out***.println(res);

}

}

public class Rectangle\_with\_classes\_and\_object {

public static void main(String[] args) {

read\_calculate\_display display = new read\_calculate\_display();

display.c();

}

}

# OUTPUT

package lab07\_2\_25;

import java.util.Scanner;

class read\_calculate\_display {

public double r() {

Scanner s = new Scanner(System.***in***);

System.***out***.println("Enter a length in cm:");

double len = s.nextInt();

System.***out***.println("Enter a breath in cm:");

double bre = s.nextInt();

s.close();

return len \* bre;

}

public void c() {

double res = r();

System.***out***.println(res);

}

}

public class Rectangle\_with\_classes\_and\_object {

public static void main(String[] args) {

read\_calculate\_display display = new read\_calculate\_display();

display.c();

}

}

# OUTPUT

Enter a length in cm:

4

Enter a breath in cm:

6

24.0

Title: Programs using classes and objects

Theory: Programs using classes and objects in Java are used to represent real-world entities with properties and behaviors. In the example below, a class named Books is created with attributes like brand, pages, and price. It also includes methods to set values (arrayofbooks) and display them (display). Objects of the Books class can be created in the main method to represent different books. This program shows how Java uses classes and objects to organize data and functionality in a structured way.

# INPUT

package lab14\_2\_25;

import java.util.Scanner;

class Books{

String brand;

int pages;

int price;

public void arrayofbooks(String b, int p, int pr) {

brand = b;

pages = p;

price = pr;

}

public void display() {

System.***out***.println("Brand:"+brand+"\nPages: "+pages+"\nPrice "+price);

}

}

public class Class\_book {

public static void main(String[] args) {

Books b1 = new Books();

b1.arrayofbooks("Penguin", 300, 450);

b1.display();

}

}

# OUTPUT

Brand:Penguin

Pages: 300

Price 450

# INPUT

package lab14\_2\_25;

import java.util.Scanner;

class Dogs {

static String *Dog\_breed*;

static String *Dog\_size*;

static int *Dog\_age*;

static void breed() {

Scanner s = new Scanner(System.***in***);

System.***out***.println("Which breed is it?");

*Dog\_breed* = s.nextLine();

s.close();

}

static void size() {

Scanner s = new Scanner(System.***in***);

System.***out***.println("How big is it: small, medium, or large?");

*Dog\_size* = s.nextLine();

s.close();

}

static void age() {

Scanner s = new Scanner(System.***in***);

System.***out***.println("What's its age?");

*Dog\_age* = s.nextInt();

s.close();

}

static void display() {

System.***out***.println("Breed of Dog is " + *Dog\_breed*);

System.***out***.println("Size of Dog is " + *Dog\_size*);

System.***out***.println("Age of Dog is " + *Dog\_age*);

}

}

public class Dog {

public static void main(String[] args) {

// **TODO** Auto-generated method stub

for(int i=0;i<2;i++) {

Dogs.*breed*();

Dogs.*size*();

Dogs.*age*();

Dogs.*display*();

}

}

}

# OUTPUT

Which breed is it?

Pug

How big is it: small, medium, or large?

small

What's its age?

4

Breed of Dog is Pug

Size of Dog is small

Age of Dog is 4

Which breed is it?

Bulldog

How big is it: small, medium, or large?

medium

What's its age?

12

Breed of Dog is Bulldog

Size of Dog is medium

Age of Dog is 12

Title: Constructors in Java

Theory: Constructors in Java are special methods used to initialize objects when they are created. They have the same name as the class and do not have a return type—not even void. Java provides a default constructor (with no parameters) if none is defined, but you can also create parameterized constructors to initialize objects with specific values. Constructors help set initial values for object attributes at the time of object creation, making code more efficient and readable. You can also overload constructors by defining multiple constructors with different parameter lists

# INPUT

package lab21\_2\_25;

import java.util.Scanner;

public class Bank\_acc {

private int accNo;

private String HoldName;

private int balance;

void deposit(int m) {

System.***out***.println("Deposit Amount: "+m);

balance+=m;

}

void withdraw(int take) {

System.***out***.println("Withdraw Amount: "+take);

balance-=take;

}

void display() {

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

}

public static void main(String[] args) {

Scanner s = new Scanner(System.***in***);

Bank\_acc Ac = new Bank\_acc();

System.***out***.print("Enter Account Number: ");

Ac.accNo = s.nextInt();

System.***out***.print("\nEnter Account Holder Name: ");

Ac.HoldName = s.next();

System.***out***.print("\nEnter Initial Balance\n");

Ac.balance = s.nextInt();

System.***out***.print("\nEnter deposit Amount\n");

Ac.deposit(s.nextInt());

Ac.display();

System.***out***.print("\nEnter Withdraw Amount\n");

Ac.withdraw(s.nextInt());

Ac.display();

s.close();

}

}

# OUTPUT

Enter Account Number: 2

Enter Account Holder Name: CHE

Enter Initial Balance

2000

Enter deposit Amount

1000

Deposit Amount: 1000

Updated Balance: 3000

Enter Withdraw Amount

2000

Withdraw Amount: 2000

Updated Balance: 1000

# INPUT

package lab21\_2\_25;

public class Car\_class {

private String brand;

private String model;

private int price;

void SD(String b,String m, int p) {

brand = b;

model = m;

price = p;

}

void Display(int count) {

System.***out***.println("Car "+count+" :Brand - "+brand+" , Model - "+model+" , Price - "+price);

}

public static void main(String[] args) {

// **TODO** Auto-generated method stub

Car\_class o = new Car\_class();

Car\_class o1 = new Car\_class();

o.SD("Toyota", "Foutuner", 5000000);

o1.SD("Honda", "Civic", 2500000);

o.Display(1);

o1.Display(2);

}

}

# OUTPUT

Car 1 :Brand - Toyota , Model - Foutuner , Price - 5000000

Car 2 :Brand - Honda , Model - Civic , Price - 2500000

# INPUT

package lab21\_2\_25;

import java.util.Scanner;

public class Stu\_rec {

String name;

int roll;

String Class;

Float percent;

Stu\_rec(){

this.name = "null";

this.roll = 0;

this.Class = "null";

this.percent = (float) 0;

}

void Read(String n,int r,String c,Float p) {

name = n;

roll = r;

Class = c;

percent = p;

}

void Display() {

System.***out***.println("\nName is : "+name);

System.***out***.println("\nRollNo. is : "+roll);

System.***out***.println("\nClass is : "+Class);

System.***out***.println("\npercentage is : "+percent+"%");

}

public static void main(String[] args) {

Scanner s = new Scanner(System.***in***);

Stu\_rec o = new Stu\_rec();

System.***out***.println("Enter name:");

String name = s.nextLine();

System.***out***.println("Enter roll number:");

int roll = s.nextInt();

s.nextLine();

System.***out***.println("Enter class:");

String className = s.nextLine();

System.***out***.println("Enter percentage:");

float percent = s.nextFloat();

o.Read(name, roll, className, percent);

o.Display();

s.close();

}

}

# Output

Enter name:

chethan

Enter roll number:

23

Enter class:

sybca

Enter percentage:

90

Name is : chethan

RollNo. is : 23

Class is : sybca

percentage is : 90.0%

Title: Intermediate programs in Java

Theory: Intermediate programs in Java go beyond the basics and help learners strengthen their understanding of object-oriented programming and core Java concepts. These programs often include the use of arrays, constructors, method overloading, file handling, exception handling, and working with classes and objects in more depth.

# INPUT

package lab28\_2\_25;

import java.util.Scanner;

public class Employee {

private String name;

private int id;

private int salary;

void SD(String n, int i, int s) {

name = n;

id = i;

if(s < 0) {

System.***out***.println("Invaild Input\n");

salary = 0;

} else {

salary = s;

}

}

void D() {

System.***out***.println("Employee Details:");

System.***out***.println("Name: "+name+"\nID: "+id+"\nSalary: "+salary);

}

public static void main(String[] args) {

Employee o = new Employee();

Scanner s = new Scanner(System.***in***);

System.***out***.println("Enter Name :");

String n = s.next();

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

int a = s.nextInt();

System.***out***.println("Enter Salary: ");

int s1 = s.nextInt();

o.SD(n, a, s1);

o.D();

s.close();

}

}

# OUTPUT

Enter Name :

S

Enter ID :

2

Enter Salary:

20000

Employee Details:

Name: S

ID: 2

Salary: 20000

# INPUT

package lab28\_2\_25;

import java.util.Scanner;

public class Library {

private String title;

private String author;

private int availableCopies;

void setdata(String t, String a, int av) {

title = t;

author = a;

availableCopies = av;

}

int Borrowbook() {

availableCopies--;

return availableCopies;

}

void display() {

System.***out***.println("Title: " + title);

System.***out***.println("Author: " + author);

System.***out***.println("Available Copies: " + availableCopies);

}

int returnbook() {

availableCopies++;

return availableCopies;

}

public static void main(String[] args) {

Scanner s = new Scanner(System.***in***);

Library[] o = new Library[3];

for (int i=0; i<3; i++) {

o[i] = new Library();

System.***out***.println("Enter Title :");

String t = s.next();

System.***out***.println("Enter Author :");

String a = s.next();

System.***out***.println("Enter No.Copies: ");

int av = s.nextInt();

o[i].setdata(t, a, av);

}

System.***out***.println("Do you want to borrow this book?");

o[1].display();

s.close();

}

}

# OUTPUT

Enter Title :

the

Enter Author :

the

Enter No.Copies:

3

Enter Title :

read

Enter Author :

read

Enter No.Copies:

10

Enter Title :

man

Enter Author :

man

Enter No.Copies:

20

Do you want to borrow this book?

Title: read

Author: read

Available Copies: 10

# INPUT

package lab28\_2\_25;

import java.util.Scanner;

public class Product {

private String name;

private int price;

private int discount;

void SD(String n, int p, int d) {

name = n;

price = p;

discount = d;

}

void D(int Dis) {

System.***out***.println("Final price after discount :"+Dis);

}

int calculateDiscount(int p, float dis) {

int disco = p;

dis /=100;

disco\*=dis;

return discount = price - disco;

}

public static void main(String[] args) {

Product o = new Product();

Scanner s = new Scanner(System.***in***);

System.***out***.println("Enter Name :");

String n = s.next();

System.***out***.println("Enter Price :");

int a = s.nextInt();

System.***out***.println("Enter Discount: ");

int s1 = s.nextInt();

o.SD(n, a, s1);

o.calculateDiscount(a, s1);

o.D(o.discount);

s.close();

}

}

# OUTPUT

Enter Name :

bottle

Enter Price :

50

Enter Discount:

25

Final price after discount :38

# INPUT

package lab28\_2\_25;

import java.util.Scanner;

public class student\_arr\_of\_objs {

private String name;

private int age;

private char grade;

void setdata(String n,int a, char g) {

name = n;

age = a;

grade = g;

}

void display(int count) {

System.***out***.println("Enter details for Student "+count+" :");

System.***out***.println("Name: "+name+"\nAge: "+age+"\nGrade: "+grade);

}

public static void main(String[] args) {

Scanner s = new Scanner(System.***in***);

student\_arr\_of\_objs[] o = new student\_arr\_of\_objs[3];

for (int i=0; i<3; i++) {

o[i] = new student\_arr\_of\_objs();

System.***out***.println("Enter Name :");

String n = s.next();

System.***out***.println("Enter Age :");

int a = s.nextInt();

System.***out***.println("Enter Grade: ");

char g = s.next().charAt(0);

o[i].setdata(n, a, g);

}

for (int i=0; i<3; i++) {

o[i].display(i + 1);

}

s.close();

}

}

# OUTPUT

Enter Name :

anish

Enter Age :

12

Enter Grade:

A

Enter Name :

faaris

Enter Age :

13

Enter Grade:

B

Enter Name :

chetan

Enter Age :

12

Enter Grade:

A

Enter details for Student 1 :

Name: anish

Age: 12

Grade: A

Enter details for Student 2 :

Name: faaris

Age: 13

Grade: B

Enter details for Student 3 :

Name: chetan

Age: 12

Grade: A

Title: inheritance and polymorphism

Theory: Inheritance allows a class to inherit properties and methods from another class. Polymorphism lets the same method behave differently based on the object, enabling flexibility and dynamic behaviour in Java programs.

# INPUT

package lab28\_3\_25;

class purse1{

private String color;

private int pocket;

private int price;

purse1 (String c, int p, int pr){

color = c;

pocket = p;

price = pr;

}

void display() {

System.***out***.println("color of purse:" + color);

System.***out***.println("No. pocket of purse:" + pocket);

System.***out***.println("price of purse:" + price);

System.***out***.println('\n');

}

}

public class purse {

public static void main(String[] args) {

purse1 o = new purse1("red", 4, 12000);

purse1 o1 = new purse1("blue", 6, 20000);

o.display();

o1.display();

}

}

# OUTPUT

color of purse:red

No. pocket of purse:4

price of purse:12000

color of purse:blue

No. pocket of purse:6

price of purse:20000

# INPUT

package lab28\_3\_25;

class Bag1 {

private String color;

private int pockets;

private int price;

Bag1(String c, int p, int pr) {

color = c;

pockets = p;

price = pr;

}

void display() {

System.***out***.println("Color of bag: " + color);

System.***out***.println("No. of pockets: " + pockets);

System.***out***.println("Price of bag: " + price);

System.***out***.println();

}

}

public class Bag {

public static void main(String[] args) {

Bag1 b1 = new Bag1("Black", 5, 2500);

Bag1 b2 = new Bag1("Grey", 7, 3500);

b1.display();

b2.display();

}

}

# OUTPUT

Color of bag: Black

No. of pockets: 5

Price of bag: 2500

Color of bag: Grey

No. of pockets: 7

Price of bag: 3500

# INPUT

package lab28\_3\_25;

class Vehicle {

void start() {

System.***out***.println("Vehicle is starting...");

}

}

class Car extends Vehicle {

void drive() {

System.***out***.println("Car is driving...");

}

}

class Bike extends Vehicle {

void ride() {

System.***out***.println("Bike is riding...");

}

}

public class Inheritance {

public static void main(String[] args) {

Car c = new Car();

c.start();

c.drive();

Bike b = new Bike();

b.start();

b.ride();

}

}

# OUTPUT

Vehicle is starting...

Car is driving...

Vehicle is starting...

Bike is riding...

# INPUT

package lab28\_3\_25;

class Shape {

void draw() {

System.***out***.println("Drawing a shape...");

}

}

class Circle extends Shape {

void draw() {

System.***out***.println("Drawing a circle...");

}

}

class Rectangle extends Shape {

void draw() {

System.***out***.println("Drawing a rectangle...");

}

}

public class Polymorphism {

public static void main(String[] args) {

Shape s;

s = new Circle();

s.draw();

s = new Rectangle();

s.draw();

}

}

# OUTPUT

Drawing a circle...

Drawing a rectangle...

Title: Exception handling

Theory: Exception handling in Java is a mechanism used to handle runtime errors, allowing the normal flow of the program to continue. It uses try, catch, throw, throws, and finally blocks to catch and manage exceptions. This prevents the program from crashing unexpectedly and helps in debugging. Common exceptions include ArithmeticException, NullPointerException.

# INPUT

package lab4\_4\_25;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

public class FileNotFound {

public static void main(String[] args) {

String fileName = "file does not exist";

File file = new File(fileName);

try {

FileInputStream stream = new FileInputStream(file);

} catch (FileNotFoundException e) {

e.printStackTrace();

}

}

}

# OUTPUT

java.io.FileNotFoundException: file does not exist (The system cannot find the file specified)

at java.base/java.io.FileInputStream.open0(Native Method)

at java.base/java.io.FileInputStream.open(FileInputStream.java:219)

at java.base/java.io.FileInputStream.<init>(FileInputStream.java:159)

at lab4\_4\_25/lab4\_4\_25.FileNotFound.main(FileNotFound.java:12)

# INPUT

package lab4\_4\_25;

public class Finally {

public static void main(String[] args) {

try {

int res = 20/0;

System.***out***.println(res);

} catch(ArithmeticException e) {

System.***out***.println("cannot divide by 0");

} finally {

System.***out***.println("Wow");

}

}

}

# OUTPUT

cannot divide by 0

Wow

# INPUT

package lab4\_4\_25;

public class indexOutOfBounds {

public static void main(String[] args) {

try {

int[] nums = {1,2,3};

System.***out***.println(nums[5]);

int res = 10/0;

System.***out***.println(res);

}

catch(ArrayIndexOutOfBoundsException e) {

System.***out***.println("array out of bounds");

}

catch(ArithmeticException e) {

System.***out***.println("cannot divide by zero");

}

catch(Exception e) {

System.***out***.println("Some Exception");

}

}

}

# OUTPUT

array out of bounds

# INPUT

package lab4\_4\_25;

public class NullPointer {

public static void main(String[] args) {

try {

int[] nums = null;

System.***out***.println(nums.length);

} catch (NullPointerException e) {

System.***out***.println("nums does not exist");

}

}

}

# OUTPUT

nums does not exist

# INPUT

package lab4\_4\_25;

public class StringOutOfBounds {

public static void main(String[] args) {

try {

String str = "Bruh";

System.***out***.println(str.charAt(10));

} catch(StringIndexOutOfBoundsException e) {

System.***out***.println("char not found on str");

}

}

}

# OUTPUT

char not found on str

# INPUT

package lab4\_4\_25;

public class Throw {

static void checkAge(int age) {

if(age < 18) {

throw new ArithmeticException("You cannot access this site");

}

else {

System.***out***.println("Welcome");

}

}

public static void main(String[] args) {

*checkAge*(17);

}

}

# OUTPUT

Exception in thread "main" java.lang.ArithmeticException: You cannot access this site

at lab4\_4\_25/lab4\_4\_25.Throw.checkAge(Throw.java:6)

at lab4\_4\_25/lab4\_4\_25.Throw.main(Throw.java:13)

# INPUT

package lab4\_4\_25;

public class Throws {

static void checkAge(int age) throws ArithmeticException {

if(age < 18) {

throw new ArithmeticException("You cannot access this site");

}

else {

System.***out***.println("Welcome");

}

}

public static void main(String[] args) {

*checkAge*(17);

}

}

# OUTPUT

Exception in thread "main" java.lang.ArithmeticException: You cannot access this site

at lab4\_4\_25/lab4\_4\_25.Throws.checkAge(Throws.java:6)

at lab4\_4\_25/lab4\_4\_25.Throws.main(Throws.java:13)

# INPUT

package lab4\_4\_25;

public class Try\_catch {

public static void main(String[] args) {

try {

int res = 10/0;

System.***out***.println(res);

}

catch(ArithmeticException e){

System.***out***.println("Cannot divide by 0");

}

System.***out***.println("continues");

}

}

# OUTPUT

Cannot divide by 0

continues