Create a class Vehicle. The class should have two fields-no_of_seats and no_of_wheels. Create two objects-Motorcycle and Car for this class. Your output should show the descriptions for Car and Motorcycle.

```java
class Vehicle {
        private int no_of_seats;
        private int no_of_wheels;
        void set(int s, int w) {
                no_of_seats = s;
                no_of_wheels = w;
        }
        void get() {
                System.out.println(no_of_seats + " " + no_of_wheels);
        }
}


public class Q1 {
        public static void main(String[] args) {
                Vehicle Car = new Vehicle();
                Vehicle Motorcycle = new Vehicle();

                Car.set(5, 4);
                Motorcycle.set(2, 1);
                Car.get();
                Motorcycle.get();
        }
}
```

Create a class with a method. The method has to decide whether a given year is a leap year or not.
Note- A year is a leap year if:-
• It has an extra day i.e. 366 instead of 365.
• It occurs in every 4 year e.g. 2008, 2012 are leap years.
• For every 100 years a special rule applies-1900 is not a leap
year but 2000 is a leap year. In those cases, we need to check whether it is divisible by 400 or not

```
import java.util.Scanner;

class Year {
    public static void isLeapYear(int year) {
        boolean flag = false;

        if (year % 4 == 0) {

            if (year % 100 == 0) {

                if (year % 400 == 0) {
                    flag = true;
                }
            } else {
                flag = true;
            }
        }

        if (flag) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }
    }
}

public class Q2 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Year: ");
        Year.isLeapYear(sc.nextInt());
        sc.close();
    }
}
```

Write a program to print the area of a rectangle by creating a class named 'Area' having two methods. First method named as 'setDim' takes length and breadth of rectangle as parameters and the second method named as 'getArea' returns the area of the rectangle. Length and breadth of rectangle are entered through keyboard

```
import java.util.Scanner;

class Area {

    private static int height;
    private static int width;

    public void setDim(int h, int w) {
        height = h;
        width = w;
    }

    public int getDim() {
        return height * width;
    }
}

public class Q3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int h = sc.nextInt();
        int w = sc.nextInt();

        Area rect = new Area();
        rect.setDim(h, w);
        System.out.println("Area Of Reactangle: " + rect.getDim());

        sc.close();
    }
}
```

Create a class named 'Student' with String variable 'name' and integer variable 'roll_no'. Assign the value of roll_no as '2' and that of name as "John" by creating an object of the class Student.

```java
class Student {
        String name;
        int roll_no;
}

public class Q4 {
        public static void main(String[] args) {
                Student obj = new Student();
                obj.name = "John";
                obj.roll_no = 2;

                System.out.println("Name: " + obj.name);
                System.out.println("Roll No: " + obj.roll_no);
        }
}
```

Assign and print the roll number, phone number and address of two students having names "Sam" and "John" respectively by creating two objects of class 'Student'.

```java
class Student {
    String name;
    String address;
    long phoneNumber;
    int roll_no;

    public void getDetails() {
        System.out.println(name);
        System.out.println(roll_no);
        System.out.println(phoneNumber);
        System.out.println(address);
    }

}

public class Q5 {
    public static void main(String[] args) {
        Student Sam = new Student();
        Sam.name = "Sam";
        Sam.roll_no = 1;
        Sam.address = "Jaipur";
        Sam.phoneNumber = 5453493712564L;
        Sam.getDetails();

        Student John = new Student();
        John.name = "John";
        John.roll_no = 2;
        John.address = "Same As Sam";
        John.phoneNumber = 9876543201L;
        John.getDetails();
    }
}
```

Write a program to print the area and perimeter of a triangle having sides of 3, 4 and 5 units by creating a class named 'Triangle' without any parameter in its constructor.

```java
class Triangle {
    private static int b, h, d;

    Triangle() {
        b = 3;
        h = 4;
        d = 5;
    }

    void perimeter() {
        System.out.println("Perimeter: " + (b + h + d));
    }

    void area() {
        System.out.println("Area: " + h * b / 2);
    }
}

public class Q6 {
    public static void main(String[] args) {
        Triangle tri = new Triangle();
        tri.area();
        tri.perimeter();
    }
}
```

Print the average of three numbers entered by user by creating a class named 'Average' having a method to calculate and print the average.

```
class Average {
    private static int a, b, c, avg;

    Average(int aa, int bb, int cc) {
        a = aa;
        b = bb;
        c = cc;
    }

    void calculateAvg() {
        avg = a + b + c / 2;
    }

    void printAvg() {
        System.out.println("Average: " + avg);
    }

}

public class Q8 {
    public static void main(String[] args) {
        Average obj = new Average(5, 5, 5);
        obj.calculateAvg();
        obj.printAvg();
    }
}
```

Print the sum, difference and product of two complex numbers by creating a class named 'Complex' with separate methods for each operation whose real and imaginary parts are entered by user.

```
import java.util.Scanner;

class ComplexNumber {
    public int real, imag;

    ComplexNumber(int x, int y) {
        real = x;
        imag = y;
    }
}

class Complex {
    private final ComplexNumber n1, n2;

    Complex(ComplexNumber n1, ComplexNumber n2) {
        this.n1 = n1;
        this.n2 = n2;
    }

    public void add() {
        ComplexNumber sum = new ComplexNumber(0, 0);
        sum.real = this.n1.real + this.n2.real;
        sum.imag = this.n1.imag + this.n2.imag;

        System.out.println("Sum: " + sum.real + " +i" + sum.imag);
    }

    public void difference() {
        ComplexNumber sub = new ComplexNumber(0, 0);
        sub.real = this.n1.real - this.n2.real;
        sub.imag = this.n1.imag - this.n2.imag;

        System.out.println("Difference: " + sub.real + " +i" + sub.imag);
    }

    public void multiply() {
        ComplexNumber mult = new ComplexNumber(0, 0);
        mult.real = this.n1.real * this.n2.real - this.n1.imag * this.n2.imag;
        mult.imag = this.n1.real * this.n2.imag + this.n1.imag * this.n2.real;
        System.out.println("Multiplication: " + mult.real + " +" + mult.imag + "i");
    }
}

public class Q9 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int x1 = sc.nextInt();
        int y1 = sc.nextInt();
        ComplexNumber n1 = new ComplexNumber(x1, y1);

        int x2 = sc.nextInt();
        int y2 = sc.nextInt();
        ComplexNumber n2 = new ComplexNumber(x2, y2);

        Complex c = new Complex(n1, n2);
        c.add();
        c.difference();
        c.multiply();

        sc.close();
    }
}
```

The class Movie is started below. An instance of class Movie represents a film.

- This class has the following three class variables:
- title, which is a String representing the title of the movie
- studio, which is a String representing the studio that made the

movie rating, which is a String representing the rating of the movie (i.e. PG13, R, etc)
public class Movie {
private String title;
private String studio;
private String rating; // your code goes here }

a)Write a constructor for the class Movie, which takes a String representing the title of the movie, a String representing the studio, and a String representing the rating as its arguments, and sets the respective class variables to these values.

b)Write a second constructor for the class Movie, which takes a String representing the title of the movie and a String representing the studio as its arguments, and sets the respective class variables to these values, while the class variable rating is set to "PG".

c)Write a method get PG, which takes an array of base type Movie as its argument, and returns a new array of only those movies in the input array with a rating of "PG". You may assume the input array is full of Movie instances. The returned array need not be full.

d) Write a piece of code that creates an instance of the class Movie with the title “Casino Royale”, the studio “Eon Productions”, and the rating “PG13

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

class Movie {
        private String title;
        private String studio;
        private String rating;
        private List<Movie> pgMovies = new ArrayList<Movie>();

        Movie() {}

        Movie(String title, String studio, String rating) {
                this.title = title;
                this.studio = studio;
                this.rating = rating;
        }

        Movie(String title, String studio) {
                this.title = title;
                this.studio = studio;
                this.rating = "PG";
        }

        public String getRating() {
                return rating;
        }

        public String getStudio() {
                return studio;
        }

        public String getTitle() {
                return title;
        }

        public List<Movie> getPG(Movie[] Movies) {

                for (Movie movie : Movies) {
                    if (movie.rating == "PG") {
                        pgMovies.add(movie);
                    }
                }
                return pgMovies;
        }
}

public class Q10 {
        public static void main(String[] args) {
                Movie m1 = new Movie("Casino Royale 123", "Eon Productions", "PG13");
                Movie m2 = new Movie("Casino Royale", "Eon Productions", "PG");
                Movie m3 = new Movie("Casin", "Eon");

                Movie[] MoviesArr = { m1, m2, m3 };

                List<Movie> pgMovies = new Movie().getPG(MoviesArr);

                for (Movie movie : pgMovies) {
                    System.out.println(movie.getTitle());
                }
        }
}
```

Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating object of Student class

```java
class Student {
    public String name;

    Student() {
        name = "Unknown";
    }

    Student(String name) {
        this.name = name;
    }

}

public class Q11 {
    public static void main(String[] args) {
        Student s1 = new Student();
        Student s2 = new Student("John");

        System.out.println(s1.name);
        System.out.println(s2.name);
    }
}
```

Write a program which will ask the user to enter his/her marks (out of 100). Define a method that will display grades according to the marks entered as below:

Marks	Grade
91-100	AA
81-90	AB
71-80	BB
61-70	BC
51-60	CD
41-50	DD
<=40	Fail

```java
import java.util.Scanner;

class Marks {
    int marks;
    String grade;

    Marks() {
        Scanner sc = new Scanner(System.in);
        System.out.print("Marks: ");
        marks = sc.nextInt();
        sc.close();
    }

    void grade() {
        if (marks >= 91 && marks <= 100) {
            grade = "AA";
        } else if (marks >= 81 && marks <= 90) {
            grade = "AB";
        } else if (marks >= 71 && marks <= 80) {
            grade = "BB";
        } else if (marks >= 61 && marks <= 70) {
            grade = "BC";
        } else if (marks >= 51 && marks <= 60) {
            grade = "CD";
        } else if (marks >= 41 && marks <= 50) {
            grade = "DD";
        } else {
            grade = "Fail";
        }

        System.out.println("Grade: " + grade);
    }
}
```

```java
public class Q12 {
        public static void main(String[] args) {
                new Marks().grade();
        }
}
```

Create a class named 'PrintNumber' to print various numbers of different datatypes by creating different methods with the same name 'printn' having a parameter for each datatype

```
class PrintNumber {
    PrintNumber(int n) {
        System.out.println("Integer: " + n);
    }

    PrintNumber(String s) {
        System.out.println("String: " + s);
    }

    PrintNumber(char c) {
        System.out.println("Character: " + c);
    }

    PrintNumber(float f) {
        System.out.println("Float: " + f);
    }

    PrintNumber(double d) {
        System.out.println("Double: " + d);
    }

    PrintNumber(int[] arr) {
        System.out.print("Array: ");
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
}

public class Q13 {
    public static void main(String[] args) {
        new PrintNumber(2);
        new PrintNumber(2.0F);
        new PrintNumber(22345.234D);
        new PrintNumber("John");

        int[] arr = {1, 2, 3, 4, 10};
        new PrintNumber(arr);
    }
}
```

Create a class to print an integer and a character with two methods having the same name but different sequence of the integer and the character parameters. For example, if the parameters of the first method are of the form (int n, char c), then that of the second method will be of the form (char c, int n).

```java
public class Q14 {
        public static void ionc(int i, char c) {
                System.out.println("Integer: " + i);
                System.out.println("Character: " + c);
        }

        public static void ionc(char c, int i) {
                System.out.println("Integer: " + i);
                System.out.println("Character: " + c);
        }

        public static void main(String[] args) {
                ionc(1, 'd');
                ionc('e', 2);
        }
}
```

Create a class to print the area of a square and a rectangle. The class has two methods with the same name but different number of parameters. The method for printing area of rectangle has two parameters which are length and breadth respectively while the other method for printing area of square has one parameter which is side of square

```
class Area {
        public void dims(int b, int l) {
                System.out.println("Araa Of Reactangle: " + (b * l));
        }

        public void dims(int side) {
                System.out.println("Araa Of Square: " + (side * side));
        }
}

public class Q15 {
        public static void main(String[] args) {
                new Area().dims(2, 5);
                new Area().dims(2);
        }
}
```

Create a class 'Student' with three data members which are name, age and address. The constructor of the class assigns default values name as "unknown", age as '0' and address as "not available". It has two members with the same name 'setInfo'. First method has two parameters for name and age and assigns the same whereas the second method takes has three parameters which are assigned to name, age and address respectively. Print the name, age and address of 10 students. Hint - Use array of objects

```
class Student {
private String name, address;
        private int age;

        Student() {
                this.name = "unknown";
                this.address = "not available";
                this.age = 0;
        }

        void setInfo(String name, int age) {
                this.name = name;
                this.age = age;
        }

        void setInfo(String name, String address, int age) {
                this.name = name;
                this.address = address;
                this.age = age;
        }

        void getInfo() {
                System.out.println("\nName: " + name);
                System.out.println("Address: " + address);
                System.out.println("Age: " + age);
        }
}

public class Q16 {
        public static void main(String[] args) {
                Student[] students = new Student[10];
                students[0] = new Student();
                students[1] = new Student();
                students[2] = new Student();
                students[3] = new Student();
                students[4] = new Student();
                students[5] = new Student();
                students[6] = new Student();
                students[7] = new Student();
                students[8] = new Student();
                students[9] = new Student();

                students[1].setInfo("A", 12);
                students[2].setInfo("B", 43);
                students[3].setInfo("C", "Jaipur", 43);
                students[4].setInfo("D", "US From UK", 32);
                students[5].setInfo("E", 2);
                students[6].setInfo("G", 42);
                students[7].setInfo("H", "Singapore", 12);
                students[9].setInfo("I", 22);

                for (Student student : students) {
                    student.getInfo();
                }
        }
}
```

Create a class 'Degree' having a method 'getDegree' that prints "I got a degree". It has two subclasses namely 'Undergraduate' and 'Postgraduate' each having a method with the same name that prints "I am an Undergraduate" and "I am a Postgraduate" respectively. Call the method by creating an object of each of the three classes.

```
class Degree {

        public void getDegree() {
                System.out.println("I got the degree");
        }

        static class Undergraduate {
                public void getDegree() {
                        System.out.println("I am an Undergraduate");
                }
        }

        static class Postgraduate {
                public void getDegree() {
                        System.out.println("I am a Postgraduate");
                }
        }
}

public class Q17 {
        public static void main(String[] args) {
                new Degree().getDegree();
                new Degree.Postgraduate().getDegree();
                new Degree.Undergraduate().getDegree();
        }
}
```

Create a class named 'Shape' with a method to print "This is This is shape". Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

```java
class Shape {
        public void show() {
                System.out.println("This is This is shape");
        }
}

class Rectangle extends Shape {
        public void show() {
                super.show();
                System.out.println("This is rectangular shape");
        }
        class Square {
                public void show() {
                        System.out.println("Square is a rectangle");
                }

                public Rectangle getParent() {
                        return Rectangle.this;
                }
        }

}

class Circle extends Shape {
        public void show() {
                System.out.println("This is circular shape");
        }
}

public class Q18 {
        public static void main(String[] args) {
                Rectangle r = new Rectangle();
                Rectangle.Square innerSquare = r.new Square();

                innerSquare.show();
                innerSquare.getParent().show();
        }
}
```

Create a class with a method that prints "This is parent class" and its subclass with another method that prints "This is child class". Now, create an object for each of the class and call

1 - method of parent class by object of parent class
2 - method of child class by object of child class
3 - method of parent class by object of child class

```java
class Parent {
        public void say() {
                System.out.println("This is parent class");
        }
}

class Child extends Parent {
        public void say() {
                System.out.println("This is child class");
        }

        public void say(boolean parent) {
                super.say();
        }
}

public class Q19 {
        public static void main(String[] args) {
                Parent p = new Parent();
                Child c = new Child();

                p.say();
                c.say();
                c.say(true);
        }
}
```

Create a class named 'Rectangle' with two data members 'length' and 'breadth' and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class 'Square' inherit the 'Rectangle' class with its constructor having a parameter for its side (suppose s) calling the constructor of its parent class as 'super(s,s)'. Print the area and perimeter of a rectangle and a square

```java
import java.util.Scanner;

class Rectangle {
        private int length, breadth;

        Rectangle(int length, int breadth) {
                this.length = length;
                this.breadth = breadth;
        }

        public void area() {
                System.out.println("Area: " + (length * breadth));
        }

        public void perimeter() {
                System.out.println("Perimeter: " + 2 * (length + breadth));
        }
}

class Square extends Rectangle {
        Square(int side) {
                super(side, side);
        }
}

public class Q20 {
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                Rectangle r = new Rectangle(10, 5);
                r.area();
                r.perimeter();

                Square s = new Square(5);
                s.area();
                s.perimeter();
                sc.close();
        }
}
```

Create a class named 'Member' having the following members:
Data members
1 – Name
2 – Age
3 - Phone number
4 – Address
5 – Salary
It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

```
class Member {
        public String name, address;
        public int phoneNumber, age, salary;

        public void printSalary() {
                System.out.println("Salary: " + salary);
        }
}

class Employee extends Member {
        public String specialization, department;
}

class Manager extends Member {
        public String specialization, department;
}

public class Q21 {
        public static void main(String[] args) {
                Employee employee = new Employee();
                employee.name = "John";
                employee.address = "Rajasthan";
                employee.age = 23;
                employee.phoneNumber = 123456;
                employee.salary = 654334;
                employee.specialization = "Pentest";
                employee.department = "Security";

                Manager manager = new Manager();
                manager.name = "Sam";
                manager.address = "India";
                manager.age = 43;
                manager.phoneNumber = 543132;
                manager.salary = 3242344;
                manager.department = "Security";
                manager.specialization = "Sr. Pentester";

                System.out.println("Manager Name: " + manager.name);
                System.out.println("Manager Age: " + manager.age);
                System.out.println("Manager Address: " + manager.address);
                System.out.println("Manager Salary: " + manager.salary);
                System.out.println("Manager Specialization: " +
        manager.specialization);
                System.out.println("Manager Department: " + manager.department);
                System.out.println("Manager Phone Number: " +
        manager.phoneNumber);

                System.out.println("\nEmployee Name: " + employee.name);
                System.out.println("Employee Age: " + employee.age);
                System.out.println("Employee Address: " + employee.address);
                System.out.println("Employee Salary: " + employee.salary);
                System.out.println("Employee Specialization: " +
        employee.specialization);
                System.out.println("Employee Department: " + employee.department);
                System.out.println("Employee Phone Number: " +
        employee.phoneNumber);
        }
}
```

An abstract class has a construtor which prints "This is constructor of abstract class", an abstract method named 'a_method' and a non-abstract method which prints "This is a normal method of abstract class". A class 'SubClass' inherits the abstract class and has a method named 'a_method' which prints "This is abstract method". Now create an object of 'SubClass' and call the abstract method and the non-abstract method.
(Analyse the result)

```
abstract class AbsClass {

    AbsClass() {
        System.out.println("This is constructor of abstract class");
    }

    abstract public void a_method();

    public void non_abstract() {
        System.out.println("This is a normal method of abstract class");
    }

}

class SubClass extends AbsClass {
    public void a_method() {
        System.out.println("This is abstract method");
    }

}

public class Q22 {
    public static void main(String[] args) {
        SubClass sb = new SubClass();
        sb.a_method();
        sb.non_abstract();
    }
}
```

Create an abstract class 'Animals' with two abstract methods 'cats' and 'dogs'. Now create a class 'Cats' with a method 'cats' which prints "Cats meow" and a class 'Dogs' with a method 'dogs' which prints "Dogs bark", both inheriting the class 'Animals'. Now create an object for each of the subclasses and call their respective methods.

```java
abstract class Animals {
        abstract void cats();
        abstract void dogs();
}

class Cats extends Animals {
        public void cats() {
                System.out.println("Cats meow");
        }

        public void dogs() {
        }
}

class Dogs extends Animals {
        void cats() {
        }

        void dogs() {
                System.out.println("Dogs bark");
        }

}

public class Q23 {
        public static void main(String[] args) {
                Cats c = new Cats();
                Dogs d = new Dogs();

                d.dogs();
                c.cats();
        }
}
```

We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

```java
abstract class Shape {
        abstract void RectangleArea(int l, int b);
        abstract void SquareArea(int s);
        abstract void CircleArea(int r);
}

class Area extends Shape {
        void RectangleArea(int l, int b) {
                System.out.println("Area Of Rectangle: " + (l * b));
        }

        void SquareArea(int s) {
                System.out.println("Area Of Square: " + (s * s));
        }

        void CircleArea(int r) {
                System.out.println("Area Of Circle: " + 3.14 * (r * r));
        }

}

public class Q24 {
        public static void main(String[] args) {
                Area area = new Area();
                area.RectangleArea(12, 3);
                area.CircleArea(4);
                area.SquareArea(5);
        }
}
```

Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.

```java
abstract class Parent {
        abstract void message();
}

class SubSecond extends Parent {
        public void message() {
                System.out.println("This is second subclass");
        }
}

class SubFirst extends Parent {
        public void message() {
                System.out.println("This is first subclass");
        }
}

public class Q25 {
        public static void main(String[] args) {
                new SubFirst().message();
                new SubSecond().message();
        }
}
```

Create an abstract class 'Bank' with an abstract method 'getBalance'. Rs. 1000, Rs.1500 and Rs. 2000 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

```java
abstract class Bank {
    abstract void getBalance();
}

class Bank1 extends Bank {
    void getBalance() {
        System.out.println("Rs. 1000");
    }
}

class Bank2 extends Bank {
    void getBalance() {
        System.out.println("Rs. 1500");
    }
}

class Bank3 extends Bank {
    void getBalance() {
        System.out.println("Rs. 2000");
    }
}

public class Q26 {
    public static void main(String[] args) {
        Bank1 b1 = new Bank1();
        Bank2 b2 = new Bank2();
        Bank3 b3 = new Bank3();

        b1.getBalance();
        b2.getBalance();
        b3.getBalance();
    }
}
```

Write a program in JAVA which will handle divide by zero , array index out of bound and null pointer exception using Exception handling.

```java
import java.util.Scanner;

public class Q29 {
    static int[] Array = null;

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

        System.out.print("Create An Array (y/n): ");
        String flag = sc.next();

        if (flag.equals("y")) {
            System.out.print("Create Array Of Lenght: ");
            Array = new int[sc.nextInt()];
        }

        System.out.print("A: ");
        int a = sc.nextInt();
        System.out.print("B: ");
        int b = sc.nextInt();

        System.out.print("Store At Index: ");
        int idex = sc.nextInt();

        try {
            Array[idex] = a / b;
        } catch (ArrayIndexOutOfBoundsException | ArithmeticException
        | NullPointerException e) {
            System.out.println(e.getMessage());
        } finally {
            sc.close();
        }
    }
}
```

Write a program for example of try and catch block. In this check whether the given array size is negative or not.

```
import java.util.Scanner;

public class Q30 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
            System.out.print("Array Size: ");
        try {
            int[] Array = new int[sc.nextInt()];
        } catch (NegativeArraySizeException e) {
            System.out.println("Invalid Input");
        } finally {
            sc.close();
        }
    }
}
```