**Lab Sheet 1**

1. WAP in Java to calculate the area of rectangle and triangle. (Make necessary assumptions if needed.)

[Hint: Use two classes for rectangle and triangle and one main class]

**Source code:**

**package** Lab1;

**import** java.util.Scanner;

**class** Rectangle {

**double** length;

**double** width;

**public** Rectangle(**double** length, **double** width) {

**this**.length = length;

**this**.width = width;

}

**public** **double** calculateArea() {

**return** length \* width;

}

}

**class** Triangle {

**double** base;

**double** height;

**public** Triangle(**double** base, **double** height) {

**this**.base = base;

**this**.height = height;

}

**public** **double** calculateArea() {

**return** 0.5 \* base \* height;

}

}

**public** **class** Area {

**public** **static** **void** main(String[] args) {

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

System.***out***.println("Enter the length of the rectangle:");

**double** rectLength = scanner.nextDouble();

System.***out***.println("Enter the width of the rectangle:");

**double** rectWidth = scanner.nextDouble();

Rectangle rectangle = **new** Rectangle(rectLength, rectWidth);

System.***out***.println("Enter the base of the triangle:");

**double** triBase = scanner.nextDouble();

System.***out***.println("Enter the height of the triangle:");

**double** triHeight = scanner.nextDouble();

Triangle triangle = **new** Triangle(triBase, triHeight);

**double** rectArea = rectangle.calculateArea();

System.***out***.println("Area of the rectangle: " + rectArea);

**double** triArea = triangle.calculateArea();

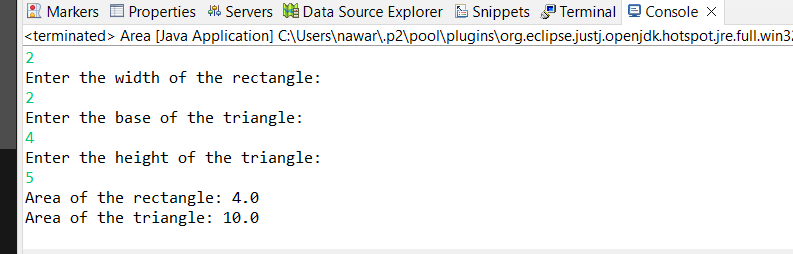
System.***out***.println("Area of the triangle: " + triArea);

scanner.close();

}

}

**Output:**



1. Implement above problem using method overloading and use constructor to initialize variables.

**Source code:**

**package** Lab1;

/////question 2

**public** **class** Overloading {

**int** area;

**public** Overloading() {

}

**public** **int** show(**int** a, **int** b) {

**return** area = a \* b;

}

**public** **int** show(**int** a, **int** b, **int** c) {

**return** area = a \* b \* c;

}

**public** **static** **void** main(String[] args) {

Overloading n1 = **new** Overloading();

n1.show(2, 4);

**int** no1 = n1.area;

System.***out***.println(no1);

n1.show(1, 2, 3);

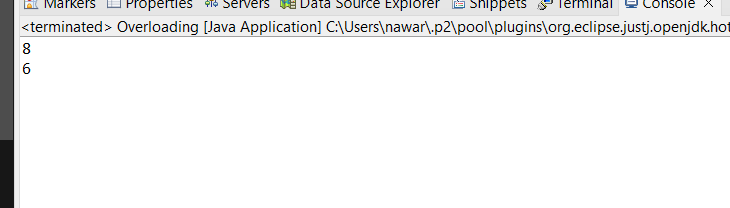
**int** no2 = n1.area;

System.***out***.println(no2);

}

}

Output:



1. WAP to find the smallest and largest element and sum of all the elements of an array.

**Source code:**

**package** Lab1;

/////question 3

**import** java.util.Scanner;

**public** **class** ArrayMinMaxSum {

**public** **static** **void** main(String[] args) {

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

System.***out***.println("Enter the size of the array:");

**int** size = scanner.nextInt();

**int**[] arr = **new** **int**[size];

System.***out***.println("Enter the elements of the array:");

**for** (**int** i = 0; i < size; i++) {

arr[i] = scanner.nextInt();

}

**int** smallest = arr[0];

**int** largest = arr[0];

**int** sum = 0;

**for** (**int** i = 0; i < size; i++) {

**if** (arr[i] < smallest) {

smallest = arr[i];

}

**if** (arr[i] > largest) {

largest = arr[i];

}

sum += arr[i];

}

System.***out***.println("Smallest Element: " + smallest);

System.***out***.println("Largest Element: " + largest);

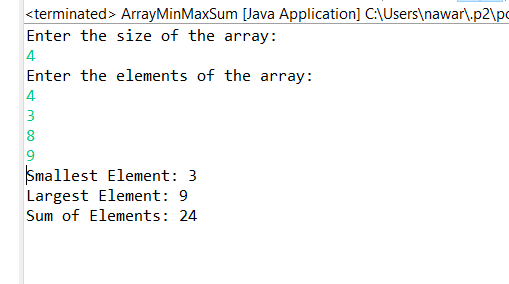
System.***out***.println("Sum of Elements: " + sum);

scanner.close();

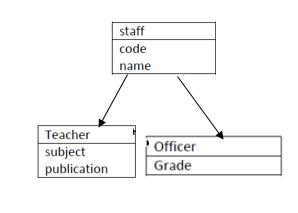
}

}

**Output:**



1. Implement the following class diagram. Make necessary assumptions if necessary.



**Source code:**

**package** Lab1;

/////question 4

**public** **class** Staff {

**public** String code;

**public** String name;

**public** **static** **class** Teacher **extends** Staff {

**public** String subject;

**public** String publication;

}

**public** **static** **class** Officer **extends** Staff {

**public** String grade;

}

**public** **static** **void** main(String[] args) {

// Teacher access

Teacher t = **new** Teacher();

t.code = "1";

t.name = "Ram";

t.subject = "Science";

t.publication = "Asia";

System.***out***.println("Teacher Information:");

System.***out***.println("Code: " + t.code);

System.***out***.println("Name: " + t.name);

System.***out***.println("Subject: " + t.subject);

System.***out***.println("Publication: " + t.publication);

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

// Officer access

Officer o = **new** Officer();

o.code = "2";

o.name = "Sita";

o.grade = "A++";

System.***out***.println("Officer Information:");

System.***out***.println("Code: " + o.code);

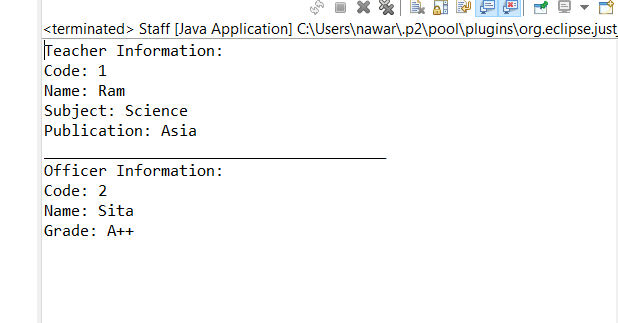
System.***out***.println("Name: " + o.name);

System.***out***.println("Grade: " + o.grade);

}

}

**Output:**



1. WAP in java to implement try-catch, finally and throw statement.

**Source code:**

**package** Lab1;

////question 5

**import** java.util.Scanner;

**public** **class** EventHandle {

**public** **static** **void** main(String[] args) {

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

**int** a= **new** Scanner(System.***in***).nextInt();

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

**int** b= **new** Scanner(System.***in***).nextInt();

**int** c;

**try** {

c= a/b;

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

**throw** **new** Exception("Done");

}**catch**(Exception e) {

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

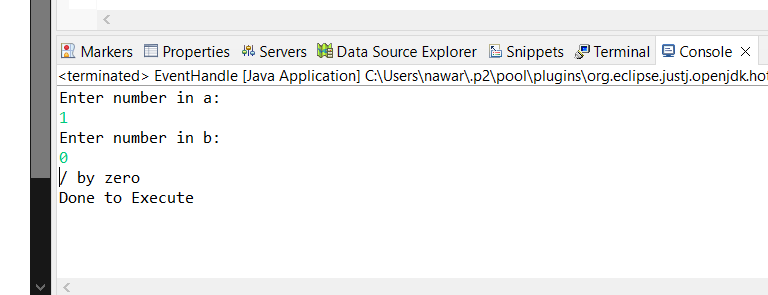
}**finally** {

System.***out***.println("Done to Execute");

}

}}

Output:



1. WAP in java to demonstrate ArithmeticException, ArrayIndexOutOfBoundException, NullPointerException, NumberFormatException separately. [Make necessary assumptions]

**Source code:**

**package** Lab1;

//question 6

**import** java.util.Scanner;

**public** **class** AllEventHandle {

**public** **static** **void** main(String[] args) {

**try** {

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

String userInput = **new** Scanner(System.***in***).nextLine();

**int** number = Integer.*parseInt*(userInput);

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

} **catch** (NumberFormatException e) {

System.***out***.println("NumberFormatException caught: " + e.getMessage());

}

**try** {

String str = **null**;

**int** length = str.length();

} **catch** (NullPointerException e) {

System.***out***.println("NullPointerException caught: " + e.getMessage());

}

**try** {

**int**[] array = **new** **int**[3];

**int** value = array[5];

} **catch** (ArrayIndexOutOfBoundsException e) {

System.***out***.println("ArrayIndexOutOfBoundsException caught: " + e.getMessage());

}

**try** {

**int** n1 = 4 / 0;

} **catch** (ArithmeticException e) {

System.***out***.println("ArithmeticException: " + e.getMessage());

}

}

}

**Output:**

