1. Working with java.lang.Boolean

b. Declare a method-local variable `status` of type `boolean` with the value `true` and convert it to a `String` using the `toString` method. (Hint: Use `Boolean.toString(Boolean)`).

Code:

```
public class programb {
     public static void main(String[] args) {
          boolean status = true;

          String s = Boolean.toString(status);

          System.out.println(s);
     }
}
```

c. Declare a method-local variable `strStatus` of type `String` with the value `"true"` and convert it to a `boolean` using the `parseBoolean` method. (Hint: Use `Boolean.parseBoolean(String)`).

```
public class ProgramC {
    public static void main(String args[]) {
        String strStatus = "true";

        boolean b = Boolean.parseBoolean(strStatus);

        System.out.println(b);
    }
}
```

d. Declare a method-local variable `strStatus` of type `String` with the value `"1"` or `"0"` and attempt to convert it to a `boolean`. (Hint: `parseBoolean` method will not work as expected with `"1"` or `"0"`).

```
public class Programd {
    public static void main(String args[]) {
        String strstatus = "1";// or "0"
        boolean status;
        if (strstatus.equals("1")) {
            status = true;
        }
        else {
            status = false;
        }
        System.out.println(status);
    }
}
```

e. Declare a method-local variable `status` of type `boolean` with the value `true` and convert it to the corresponding wrapper class using `Boolean.valueOf()`. (Hint: Use `Boolean.valueOf(boolean)`).

```
public class ProgramE {
    public static void main(String args[]) {
        boolean status = true;
        Boolean b = Boolean.valueOf(status);

        System.out.println(b);
    }
}
```

f. Declare a method-local variable `strStatus` of type `String` with the value `"true"` and convert it to the corresponding wrapper class using `Boolean.valueOf()`. (Hint: Use `Boolean.valueOf(String)`).

```
public class ProgramF {
    public static void main(String args[]) {
        String strstatus = "true";

        Boolean b = Boolean.valueOf(strstatus);

        System.out.println(b);
    }
}
```

2. Working with java.lang.Byte

b. Write a program to test how many bytes are used to represent a byte value using the BYTES field. (Hint: Use Byte.BYTES).

```
public class Programb {
    public static void main(String args[]) {
        int byteSize = Byte.BYTES;

        System.out.println(byteSize);
    }
}
```

c. Write a program to find the minimum and maximum values of byte using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Byte.MIN_VALUE and Byte.MAX_VALUE).

```
public class PorgramC {
    public static void main(String[] args) {
        byte min_val = (Byte.MIN_VALUE);
        byte max_val = (Byte.MAX_VALUE);

        System.out.println(min_val);
        System.out.println(max_val);
    }
}
```

d. Declare a method-local variable number of type byte with some value and convert it to a String using the toString method. (Hint: Use Byte.toString(byte)).

```
public class ProgramD {
    public static void main(String args[]) {
        byte number = 1;
        String s = Byte.toString(number);
        System.out.println(s);
    }
}
```

e. Declare a method-local variable strNumber of type String with some value and convert it to a byte value using the parseByte method. (Hint: Use Byte.parseByte(String)).

```
public class ProgramE {
    public static void main(String[] args) {
        String strnum = "55";
        byte b = Byte.parseByte(strnum);
        System.out.println(b);
    }
}
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a byte value. (Hint: parseByte method will throw a NumberFormatException).

```
public class ProgramF {
    public static void main(String args[]) {
        String num = "Ab12Cd3";

        byte b = Byte.parseByte(num);
        System.out.println(b);
    }
}
```

g. Declare a method-local variable number of type byte with some value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(byte)).

```
public class ProgramG {
    public static void main(String[] args) {
        byte number = 55;
        Byte b = Byte.valueOf(number);
        System.out.println(b);
    }
}
```

h. Declare a method-local variable strNumber of type String with some byte value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(String))

```
public class ProgramH {
    public static void main(String[] args) {
        String strnumber = "100";
        Byte b = Byte.valueOf(strnumber);
        System.out.println(b);
    }
}
```

3. Working with java.lang.Short

b. Write a program to test how many bytes are used to represent a short value using the BYTES field. (Hint: Use Short.BYTES).

```
public class Programb {
    public static void main(String[] args) {
        int s = Short.BYTES;
        System.out.println(s);
    }
}
```

c. Write a program to find the minimum and maximum values of short using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Short.MIN_VALUE and Short.MAX_VALUE).

```
public class ProgramC {
    public static void main(String[] args) {
        short min_value = Short.MIN_VALUE;
        short max_value = Short.MAX_VALUE;

        System.out.println(min_value);
        System.out.println(max_value);
    }
}
```

d. Declare a method-local variable number of type short with some value and convert it to a String using the toString method. (Hint: Use Short.toString(short)).

```
public class ProgramD {
    public static void main(String[] args) {
        short number = 1000;
        String s = Short.toString(number);
        System.out.println(s);
    }
}
```

e. Declare a method-local variable strNumber of type String with some value and convert it to a short value using the parseShort method. (Hint: Use Short.parseShort(String)).

```
public class ProgramE {
    public static void main(String[] args) {
        String strnum = "1000";
        short s = Short.parseShort(strnum);
        System.out.println(s);
    }
}
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a short value. (Hint: parseShort method will throw a NumberFormatException).

```
public class ProgramF {
    public static void main(String[] args) {
        String strnum = "Ab12Cd3";
        short s = Short.parseShort(strnum);
        System.out.println(s);
    }
}
```

g. Declare a method-local variable number of type short with some value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(short)).

```
public class ProgramG {
    public static void main(String[] args) {
        short number = 15000;
        Short s = Short.valueOf(number);
        System.out.println(s);
    }
}
```

h. Declare a method-local variable strNumber of type String with some short value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(String)).

```
public class ProgramH {
    public static void main(String[] args) {
        String strnum = "1500";
        Short s = Short.valueOf(strnum);
        System.out.println(s);
    }
}
```

4. Working with java.lang.Integer

b. Write a program to test how many bytes are used to represent an int value using the BYTES field. (Hint: Use Integer.BYTES).

```
public class Programb {
    public static void main(String[] args) {
        int i = Integer.BYTES;
        System.out.println(i);
    }
}
```

c. Write a program to find the minimum and maximum values of int using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Integer.MIN_VALUE and Integer.MAX_VALUE).

```
public class ProgramC {
    public static void main(String[] args) {
        int min_value = Integer.MIN_VALUE;
        int max_value = Integer.MAX_VALUE;

        System.out.println(min_value);
        System.out.println(max_value);
    }
}
```

d. Declare a method-local variable number of type int with some value and convert it to a String using the toString method. (Hint: Use Integer.toString(int)).

```
public class ProgramD {
    public static void main(String[] args) {
        int number = 10000;
        String s = Integer.toString(number);
        System.out.println(s);
    }
}
```

e. Declare a method-local variable strNumber of type String with some value and convert it to an int value using the parseInt method. (Hint: Use Integer.parseInt(String)).

```
public class ProgramE {
    public static void main(String[] args) {
        String strnumber = "10050";
        int number = Integer.parseInt(strnumber);
        System.out.println(number);
    }
}
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to an int value. (Hint: parseInt method will throw a NumberFormatException).

```
public class ProgramF {
    public static void main(String[] args) {
        String strnum = "Ab12Cd3";
        int num = Integer.parseInt(strnum);
        System.out.println(num);
    }
}
```

g. Declare a method-local variable number of type int with some value and convert it to the corresponding wrapper class using Integer.valueOf(). (Hint: Use Integer.valueOf(int)).

```
public class ProgramG {
    public static void main(String[] args) {
        int number = 500;
        Integer i = Integer.valueOf(number);
        System.out.println(i);
    }
}
```

h. Declare a method-local variable strNumber of type String with some integer value and convert it to the corresponding wrapper class using Integer.valueOf(). (Hint: Use Integer.valueOf(String)).

```
public class ProgramH {
    public static void main(String[] args) {
        String strnumber = "1000";
        Integer i = Integer.valueOf(strnumber);
        System.out.println(i);
    }
}
```

i. Declare two integer variables with values 10 and 20, and add them using a method from the Integer class. (Hint: Use Integer.sum(int, int)).

```
public class Programi {
    public static void main(String[] args) {
    int num1 = 10;
    int num2 = 20;
    int sum = Integer.sum(num1, num2);
    System.out.println(sum);
    }
}
```

j. Declare two integer variables with values 10 and 20, and find the minimum and maximum values using the Integer class. (Hint: Use Integer.min(int, int) and Integer.max(int, int)).

```
public class ProgramJ {
    public static void main(String[] args) {
        int num1 = 10;
        int num2 = 20;
        int min = Integer.min(num1, num2);
        int max = Integer.max(num1, num2);
        System.out.println("Min is " + min);
        System.out.println("Max is " + max);
    }
}
```

k. Declare an integer variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Integer class. (Hint: Use Integer.toBinaryString(int), Integer.toOctalString(int), and Integer.toHexString(int)).

```
public class ProgramK {
    public static void main(String[] args) {
        int num = 7;
        String b = Integer.toBinaryString(num);
        String o = Integer.toOctalString(num);
        String d = Integer.toHexString(num);

        System.out.println(b);
        System.out.println(o);
        System.out.println(d);
    }
}
```

5. Working with java.lang.Long

b. Write a program to test how many bytes are used to represent a long value using the BYTES field. (Hint: Use Long.BYTES).

```
public class Programb {
    public static void main(String[] args) {
        int num = Long.BYTES;
        System.out.println(num);
    }
}
```

c. Write a program to find the minimum and maximum values of long using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Long.MIN_VALUE and Long.MAX_VALUE).

```
public class ProgramC {
    public static void main(String[] args) {
        long min_value = Long.MIN_VALUE;
        long max_value = Long.MAX_VALUE;

        System.out.println(min_value);
        System.out.println(max_value);
    }
}
```

d. Declare a method-local variable number of type long with some value and convert it to a String using the toString method. (Hint: Use Long.toString(long)).

```
public class ProgramD {
    public static void main(String[] args) {
        long number = 500;
        String s = Long.toString(number);
        System.out.println(s);
    }
}
```

e. Declare a method-local variable strNumber of type String with some value and convert it to a long value using the parseLong method. (Hint: Use Long.parseLong(String)).

```
public class ProgrameE {
    public static void main(String[] args) {
        String strnum = "5000";
        long l = Long.parseLong(strnum);
        System.out.println(l);
    }
}
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a long value. (Hint: parseLong method will throw a NumberFormatException).

```
public class ProgramF {
    public static void main(String[] args) {
        String strnum = "Ab12Cd3";
        long l = Long.parseLong(strnum);
        System.out.println(l);
    }
}
```

g. Declare a method-local variable number of type long with some value and convert it to the corresponding wrapper class using Long.valueOf(). (Hint: Use Long.valueOf(long)).

```
public class ProgramG {
    public static void main(String[] args) {
        long number = 501;
        Long l = Long.valueOf(number);

        System.out.println(l);
    }
}
```

h. Declare a method-local variable strNumber of type String with some long value and convert it to the corresponding wrapper class using Long.valueOf(). (Hint: Use Long.valueOf(String)).

```
public class ProgramH {
    public static void main(String[] args) {
        String number = "501";

        Long l = Long.valueOf(number);

        System.out.println(l);
    }
}
```

i. Declare two long variables with values 1123 and 9845, and add them using a method from the Long class. (Hint: Use Long.sum(long, long)).

```
public class ProgramI {
    public static void main(String[] args) {
        long num1 = 1123;
        long num2 = 9845;
        long sum = Long.sum(num1, num2);
        System.out.println(sum);
    }
}
```

j. Declare two long variables with values 1122 and 5566, and find the minimum and maximum values using the Long class. (Hint: Use Long.min(long, long) and Long.max(long, long)).

```
public class ProgramJ {
    public static void main(String[] args) {
        long num1 = 1122;
        long num2 = 5566;

        long min = Long.min(num1, num2);
        long max = Long.max(num1, num2);

        System.out.println("Min is " + min);
        System.out.println("Min is " + max);
    }
}
```

k. Declare a long variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Long class. (Hint: Use Long.toBinaryString(long), Long.toOctalString(long), and Long.toHexString(long)).

```
public class ProgramK {
    public static void main(String[] args) {
        long num = 7;
        String b = Long.toBinaryString(num);
        String o = Long.toOctalString(num);
        String d = Long.toHexString(num);

        System.out.println(b);
        System.out.println(o);
        System.out.println(d);
    }
}
```

6. Working with java.lang.Float

b. Write a program to test how many bytes are used to represent a float value using the BYTES field. (Hint: Use Float.BYTES).

```
public class Programb {
    public static void main(String[] args) {
        int num = Float.BYTES;
        System.out.println(num);
    }
}
```

c. Write a program to find the minimum and maximum values of float using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Float.MIN_VALUE and Float.MAX_VALUE).

```java
public class ProgramC {
    public static void main(String[] args) {
        float min_value = Float.MIN_VALUE;
        float max_value = Float.MAX_VALUE;

        System.out.println(min_value);
        System.out.println(max_value);
    }
}
```

d. Declare a method-local variable number of type float with some value and convert it to a String using the toString method. (Hint: Use Float.toString(float)).

```java
public class ProgramD {
    public static void main(String[] args) {
        float number = 500.52f;

        String s = Float.toString(number);

        System.out.println(s);
    }
}
```

e. Declare a method-local variable strNumber of type String with some value and convert it to a float value using the parseFloat method. (Hint: Use Float.parseFloat(String)).

```java
public class ProgramE {
    public static void main(String[] args) {
        String strnum = "5000.52";

        float f = Float.parseFloat(strnum);

        System.out.println(f);
    }
}
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a float value. (Hint: parseFloat method will throw a NumberFormatException).

```java
public class ProgramF {
    public static void main(String[] args) {
        String strnum = "Ab12Cd3";

        float f = Float.parseFloat(strnum);

        System.out.println(f);
    }
}
```

g. Declare a method-local variable number of type float with some value and convert it to the corresponding wrapper class using Float.valueOf(). (Hint: Use Float.valueOf(float)).

```
public class ProgramG {
    public static void main(String[] args) {
        float number = 501.58f;

        Float f = Float.valueOf(number);

        System.out.println(f);
    }
}
```

h. Declare a method-local variable strNumber of type String with some float value and convert it to the corresponding wrapper class using Float.valueOf(). (Hint: Use Float.valueOf(String)).

```
public class ProgramH {
    public static void main(String[] args) {
        String strnumber = "501.63";

        Float f = Float.valueOf(strnumber);

        System.out.println(f);
    }
}
```

i. Declare two float variables with values 112.3 and 984.5, and add them using a method from the Float class. (Hint: Use Float.sum(float, float)).

```
public class ProgramI {
    public static void main(String[] args) {
        float num1 = 112.3f;
        float num2 = 984.5f;
        float sum =Float.sum(num1, num2);

        System.out.println(sum);
    }
}
```

j. Declare two float variables with values 112.2 and 556.6, and find the minimum and maximum values using the Float class. (Hint: Use Float.min(float, float) and Float.max(float, float)).

```
public class ProgramJ {
    public static void main(String[] args) {
        float num1 = 112.2f;
        float num2 = 556.6f;
        float min = Float.min(num1, num2);
        float max = Float.max(num1, num2);
```

```
        System.out.println("Min is " + min);
        System.out.println("Min is " + max);
    }
}
```

k. Declare a float variable with the value -25.0f. Find the square root of this value. (Hint: Use Math.sqrt() method).

```
public class ProgramK {
    public static void main(String[] args) {
        float num = -25.0f;
        float sq = (float)Math.sqrt(num);

        System.out.println(sq);
    }
}
```

l. Declare two float variables with the same value, 0.0f, and divide them. (Hint: Observe the result and any special floating-point behavior).

```
public class ProgramL {
    public static void main(String[] args) {
        float num1 = 0.0f;
        float num2 = 0.0f;
        float div = (num1/num2);
        System.out.println(div);
    }
}
//Can't divide the number 0 by 0
```

7. Working with java.lang.Double

b. Write a program to test how many bytes are used to represent a double value using the BYTES field. (Hint: Use Double.BYTES).

```
public class Programb {
    public static void main(String[] args) {
        int num = Double.BYTES;

        System.out.println(num);
    }
}
```

c. Write a program to find the minimum and maximum values of double using the MIN_VALUE and MAX_VALUE fields. (Hint: Use Double.MIN_VALUE and Double.MAX_VALUE).

```
public class ProgramC {
    public static void main(String[] args) {
        double min_value = Double.MIN_VALUE;
        double max_value = Double.MAX_VALUE;

        System.out.println(min_value);
        System.out.println(max_value);
    }
}
```

d. Declare a method-local variable number of type double with some value and convert it to a String using the toString method. (Hint: Use Double.toString(double)).

```
public class ProgramD {
    public static void main(String[] args) {
        double number = 500.558;

        String s = Double.toString(number);
        System.out.println(s);
    }
}
```

e. Declare a method-local variable strNumber of type String with some value and convert it to a double value using the parseDouble method. (Hint: Use Double.parseDouble(String)).

```
public class ProgramE {
    public static void main(String[] args) {
        String strnum = "5000.582";
        double d = Double.parseDouble(strnum);
        System.out.println(d);
    }
}
```

f. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a double value. (Hint: parseDouble method will throw a NumberFormatException).

```
public class ProgramF {
    public static void main(String[] args) {
        String strnum = "Ab12Cd3";
        double d = Double.parseDouble(strnum);
        System.out.println(d);
    }

}
```

g. Declare a method-local variable number of type double with some value and convert it to the corresponding wrapper class using Double.valueOf(). (Hint: Use Double.valueOf(double)).

```
public class ProgramG {
    public static void main(String[] args) {
        double number = 501.588;

        Double d = Double.valueOf(number);

        System.out.println(d);
    }
}
```

h. Declare a method-local variable strNumber of type String with some double value and convert it to the corresponding wrapper class using Double.valueOf(). (Hint: Use Double.valueOf(String)).

```
public class ProgramH {
    public static void main(String[] args) {
        String strnumber = "501.6368";

        Double d = Double.valueOf(strnumber);

        System.out.println(d);
    }
}
```

i. Declare two double variables with values 112.3 and 984.5, and add them using a method from the Double class. (Hint: Use Double.sum(double, double)).

```
public class Programi {
    public static void main(String[] args) {
        double num1 = 112.3;
        double num2 = 984.5;
        double sum = Double.sum(num1, num2);

        System.out.println(sum);
    }
}
```

j. Declare two double variables with values 112.2 and 556.6, and find the minimum and maximum values using the Double class. (Hint: Use Double.min(double, double) and Double.max(double, double)).

```
public class Programj {
    public static void main(String[] args) {
        double num1 = 112.2;
        double num2 = 556.6;
        double min = Double.min(num1, num2);
```

```
        double max = Double.max(num1, num2);

        System.out.println("Min is " + min);
        System.out.println("Min is " + max);
    }
}
```

k. Declare a double variable with the value -25.0. Find the square root of this value. (Hint: Use Math.sqrt() method).

```
public class Programk {
    public static void main(String[] args) {
        double number = -25.0;
        double sq = Math.sqrt(number);

        System.out.println(sq);
    }
}
//Cannot find square of negative number
```

l. Declare two double variables with the same value, 0.0, and divide them. (Hint: Observe the result and any special floating-point behavior).

```
public class ProgramL {
    public static void main(String[] args) {
        double num1 = 0.0;
        double num2 = 0.0;
        double div = (num1/num2);
        System.out.println(div);
    }
}
//Can't divide the number 0 by 0
```

8. Conversion between Primitive Types and Strings

Initialize a variable of each primitive type with a user-defined value and convert it into String:

- First, use the toString method of the corresponding wrapper class. (e.g., Integer.toString()).
- Then, use the valueOf method of the String class. (e.g., String.valueOf()).

```
public class Program8 {

    public static void main(String[] args) {
        byte byteValue = 10;
        short shortValue = 3000;
        int intValue = 123456;
        long longValue = 9876543210L;
```

```
        float floatValue = 12.34f;
        double doubleValue = 567.89;
        char charValue = 'A';
        boolean boolValue = true;

        String byteString = Byte.toString(byteValue);
        String shortString = Short.toString(shortValue);
        String intString = Integer.toString(intValue);
        String longString = Long.toString(longValue);
        String floatString = Float.toString(floatValue);
        String doubleString = Double.toString(doubleValue);
        String charString = Character.toString(charValue);
        String boolString = Boolean.toString(boolValue);

        String byteString2 = String.valueOf(byteValue);
        String shortString2 = String.valueOf(shortValue);
        String intString2 = String.valueOf(intValue);
        String longString2 = String.valueOf(longValue);
        String floatString2 = String.valueOf(floatValue);
        String doubleString2 = String.valueOf(doubleValue);
        String charString2 = String.valueOf(charValue);
        String boolString2 = String.valueOf(boolValue);

        System.out.println("Using toString methods:");
        System.out.println("Byte: " + byteString);
        System.out.println("Short: " + shortString);
        System.out.println("Int: " + intString);
        System.out.println("Long: " + longString);
        System.out.println("Float: " + floatString);
        System.out.println("Double: " + doubleString);
        System.out.println("Char: " + charString);
        System.out.println("Boolean: " + boolString);

        System.out.println("\nUsing String.valueOf methods:");
        System.out.println("Byte: " + byteString2);
        System.out.println("Short: " + shortString2);
        System.out.println("Int: " + intString2);
        System.out.println("Long: " + longString2);
        System.out.println("Float: " + floatString2);
        System.out.println("Double: " + doubleString2);
        System.out.println("Char: " + charString2);
        System.out.println("Boolean: " + boolString2);
    }
}
```

9. Default Values of Primitive Types

Declare variables of each primitive type as fields of a class and check their default values. (Note: Default values depend on whether the variables are instance variables or static variables).

```java
public class Program9 {

    byte byteValue;
    short shortValue;
    int intValue;
    long longValue;
    float floatValue;
    double doubleValue;
    char charValue;
    boolean boolValue;

    static byte staticByteValue;
    static short staticShortValue;
    static int staticIntValue;
    static long staticLongValue;
    static float staticFloatValue;
    static double staticDoubleValue;
    static char staticCharValue;
    static boolean staticBoolValue;

    public static void main(String[] args) {
        Program9 obj = new Program9();

        System.out.println("Instance variable default values:");
        System.out.println("byte: " + obj.byteValue);
        System.out.println("short: " + obj.shortValue);
        System.out.println("int: " + obj.intValue);
        System.out.println("long: " + obj.longValue);
        System.out.println("float: " + obj.floatValue);
        System.out.println("double: " + obj.doubleValue);
        System.out.println("char: " + (int) obj.charValue);
        System.out.println("boolean: " + obj.boolValue);

        System.out.println("\nStatic variable default values:");
        System.out.println("byte: " + staticByteValue);
        System.out.println("short: " + staticShortValue);
        System.out.println("int: " + staticIntValue);
        System.out.println("long: " + staticLongValue);
        System.out.println("float: " + staticFloatValue);
        System.out.println("double: " + staticDoubleValue);
        System.out.println("char: " + (int) staticCharValue);
        System.out.println("boolean: " + staticBoolValue);
    }
}
```

10. Arithmetic Operations with Command Line Input

Write a program that accepts two integers and an arithmetic operator (+, -, *, /) from the command line. Perform the specified arithmetic operation based on the operator provided. (Hint: Use switch-case for operations).

```java
import java.util.Scanner;
public class Program10 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter First number ");
        double a = sc.nextDouble();

        System.out.print("Enter Second number ");
        double b = sc.nextDouble();

        System.out.print("Enter the operator (+, -, *, /) ");
        char operator = sc.next().charAt(0);

        double result = 0;

        switch (operator) {
            case '+':
                result = (a + b);
                break;

            case '-':
                result = (a - b);
                break;

            case '*':
                result = (a * b);
                break;

            case '/':

                if (b != 0) {
                    result = (a / b);
                } else {
                    System.out.println("Division by 0 is not allowed");
                }
                break;

            default:
                System.out.println("Operator is invalid");
        }
        System.out.println("result is: " + result);
    }
}
```