Assignment 2

1. Working with java.lang.Boolean
2. Explore the [Java API documentation for java.lang.Boolean](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html) and observe its modifiers and super types.

public: The Boolean class is publicly accessible, meaning it can be accessed from any other class.

final: This means the class cannot be subclassed, so no other

class can extend Boolean

public final class Boolean

extends Object

implements Serializable, Comparable<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) ).

Program:

package org.solution;

public class ques1b {

public static void main(String[]args) {

//String strStatus = "true" ;

boolean status = true;

String str = new String();

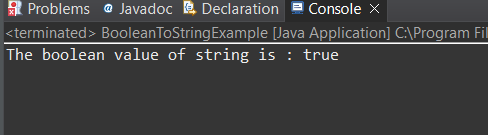
str = Boolean.*toString*(status);

//Boolean Status = Boolean.parseBoolean (strStatus);

System.***out***.println("The string value converted to boolean is: " + str);

}

}



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)).

Program:

package org.solution;

public class ques1c {

public static void main (String []args) {

String strStatus = new String("true");

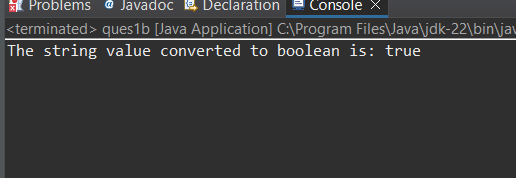
boolean Status= Boolean.*parseBoolean*(strStatus);

System.*out*.println("The string value converted to boolean is: " + Status);

}

}

Output:



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").

**Program:**

package org.solution;

public class ques1d {

public static void main(String[]args) {

String strStatus = "1";

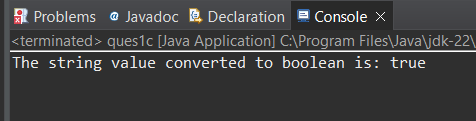
boolean B = strStatus.equals("1");

System.*out*.println(B);

}

}

Output:



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)).

Program:

package org.solution;

public class ques1e {

public static void main(String[]args) {

boolean Status = true;

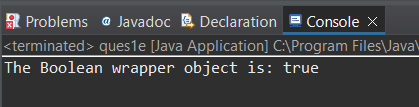
Boolean StatusWrapper = Boolean.*valueOf*(Status);

System.*out*.println("The Boolean wrapper object is: " + StatusWrapper);

}

}

Output:



**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)).

Program:

package org.solution;

public class ques1f{

public static void main(String[] args) {

String strStatus = "true”;

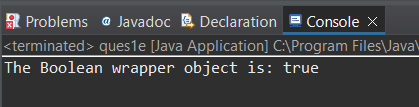
Boolean statusWrapper = Boolean.*valueOf*(strStatus);

System.*out*.println("The Boolean wrapper object is: " + statusWrapper);

}

}

Output:



**g.** Experiment with converting a boolean value into other primitive types or vice versa and observe the results.

Program:

package org.solution;

public class ques1g {

public static void main(String[] args) {

boolean B = true;

int A = (int)B;

int i = 30;

boolean X =(boolean)i;

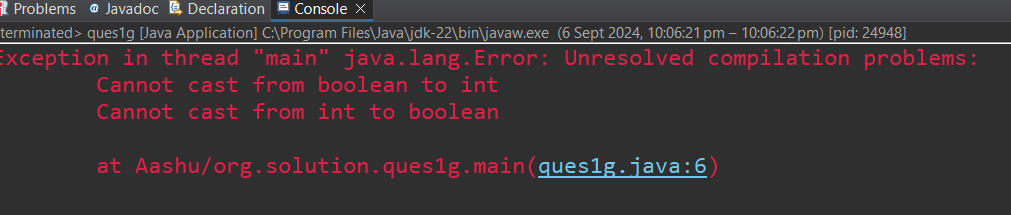
System.*out*.println(A);

System.*out*.println(X);

}

}

Output:



**2. Working with java.lang.Byte**

**a.** Explore the [Java API documentation for java.lang.Byte](https://docs.oracle.com/javase/8/docs/api/java/lang/Byte.html) and observe its modifiers and super types.

The Byte class is part of the java.lang package and is declared as:

public final class Byte extends Number implements Comparable<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).

program:

package org.solution;

public class ques2b {

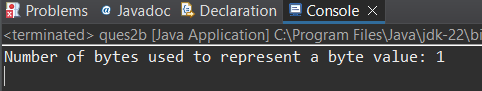
public static void main(String[]args) {

System.*out*.println("Number of bytes used to represent a byte value: " + Byte.*BYTES*);

}

}

Output:



**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).

Program:

package org.solution;

public class ques2c {

public static void main(String[]args) {

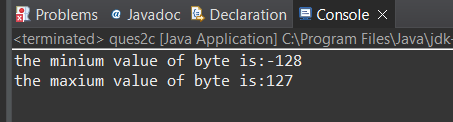
System.out.println("the minium value of byte is:" + Byte.MIN\_VALUE);

System.out.println("the maxium value of byte is:" + Byte.MAX\_VALUE);

}

}

Output:



**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)).

Program:

package org.solution;

public class ques2d {

public static void main(String[]args) {

Byte number = 10;

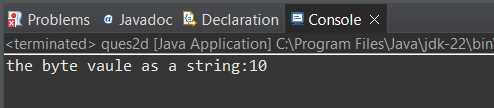
String numberString = Byte.*toString*(number);

System.***out***.println("the byte vaule as a string:" + numberString);

}

}

Output:



**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)).

Program:

package org.solution;

public class ques2e {

public static void main(String[]args) {

String strNumber = "12";

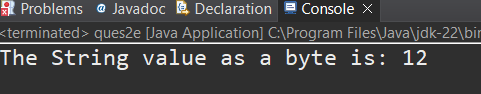
byte number = Byte.*parseByte*(strNumber);

System**.***out*.println("The String value as a byte is: " + number);

}

}

Output:



**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).

Program:

package org.solution;

public class ques2f {

public static void main(String[] args) {

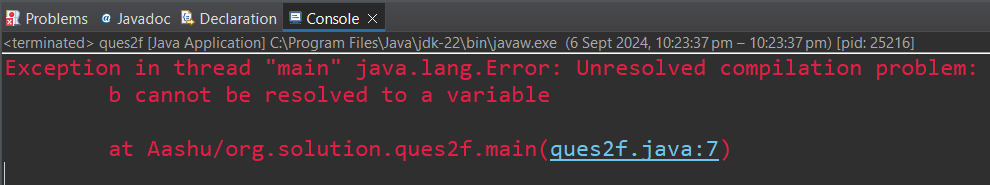
String strNumber = "Ab12Cd3";

System.***out***.println("converted to byte :"+b);

}

}

Output:



**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)).

Program:

package org.solution;

public class ques2g {

public static void main(String[] args) {

byte number = 40;

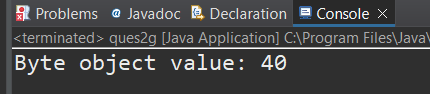
Byte B =Byte.*valueOf*(number);

System.*out*.println("Byte object value: " +B);

}

}

Output:



**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)).

Program:

package org.solution;

public class ques2h {

public static void main(String[] args) {

String strNumber = "13";

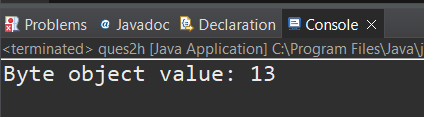
Byte B=Byte.*valueOf*(strNumber);

System.*out*.println("Byte object value: " +B);

}

}

Output:



1. Experiment with converting a byte value into other primitive types or vice versa and observe the results.

Program:

package org.solution;

public class ques2i {

public static void main(String[] args) {

byte A = 30;

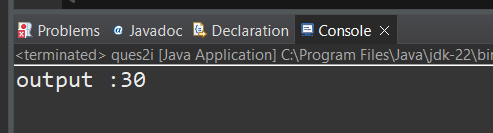
int B = (int) A;

System.out.println("output: " + B);

}

}

Output:



**3. Working with java.lang.Short**

**a.** Explore the [Java API documentation for java.lang.Short](https://docs.oracle.com/javase/8/docs/api/java/lang/Short.html) and observe its modifiers and super types.

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

program:

package org.solution;

public class ques3b {

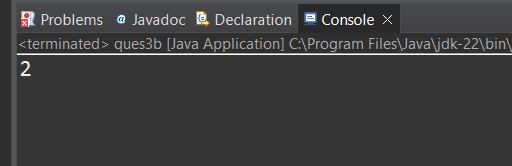
public static void main(String[] args) {

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

}

}

Output:



**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).

Program:

package org.solution;

public class ques3c {

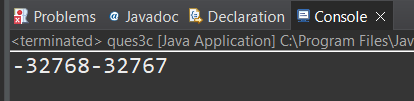
public static void main(String[]args) {

System.*out*.println(Short.*MIN\_VALUE* + "-" + Short.*MAX\_VALUE*);

}

}

Output:



**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)).

Program:

package org.solution;

public class ques3d {

public static void main(String[] args) {

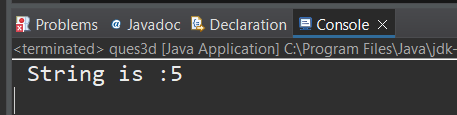
short number= 5;

System.***out***.println( " String is :"+Short.*toString*(number));

}

}

Output:



**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)).

Program:

package org.solution;

public class ques3e {

public static void main(String[] args) {

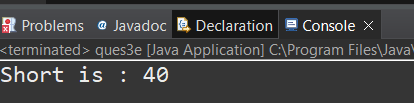
String strNumber = "40";

System.***out***.println("Short is : "+ Short.*parseShort*(strNumber));

}

}

Output:



**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).

Program:

package org.solution;

public class ques3f {

public static void main(String[] args) {

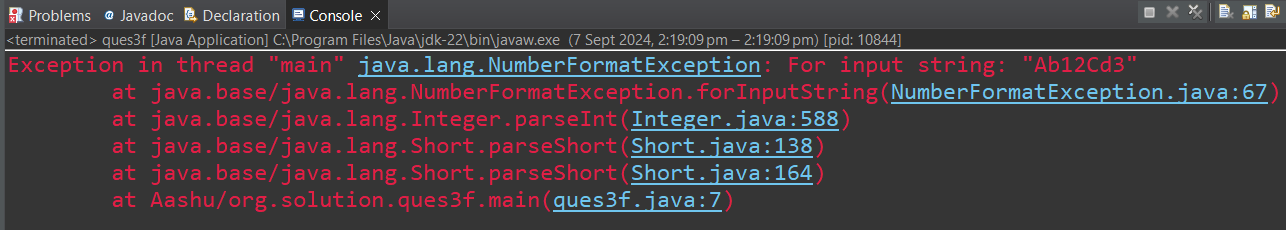
String strNumber = "Ab12Cd3";

System.***out***.println(" Short is :"+Short.*parseShort*(strNumber));

}

}

Output:



**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)).

Program:

package org.solution;

public class ques3g {

public static void main(String[] args) {

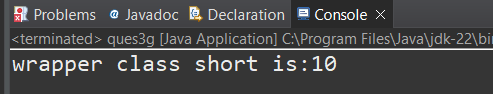
short number = 10;

System.***out***.println("wrapper class short is:" +Short.*valueOf*(number));

}

}

Output:



**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)).

Program:

package org.solution;

public class ques3h {

public static void main(String[] args) {

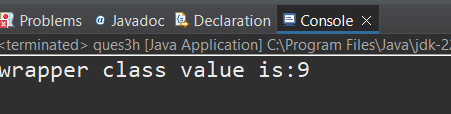
String strNumber = "9";

System.***out***.println("wrapper class value is:"+Short.*valueOf*(strNumber))

}

}

Output:



**i.** Experiment with converting a short value into other primitive types or vice versa and observe the results.

Program:

package org.solution;

public class ques3i {

public static void main(String[] args) {

short number=8;

int a=20;

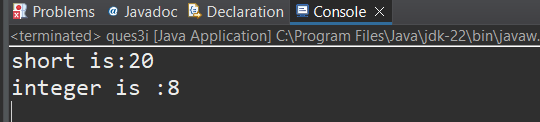
System.***out***.println("short is:"+(short)a);

System.***out***.println("integer is :"+(int)number);

}

}

Output:



**4. Working with java.lang.Integer**

**a.** Explore the [Java API documentation for java.lang.Integer](https://docs.oracle.com/javase/8/docs/api/java/lang/Integer.html) and observe its modifiers and super types

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

program:

package org.solution;

public class ques4b {

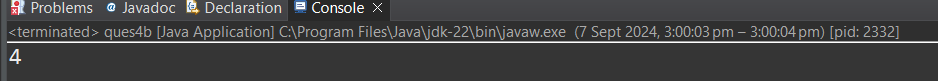
public static void main(String[]args) {

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

}

}

Output:



**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).

Program:

package org.solution;

public class ques4c {

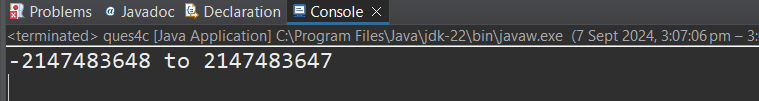
public static void main(String[] args) {

System.***out***.println(Integer.***MIN\_VALUE*** +" to "+Integer.***MAX\_VALUE***);

}

}

Output:



**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)).

Program:

package org.solution;

public class ques4d {

public static void main(String[] args) {

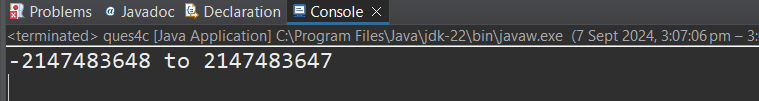
int number =45;

System.***out***.println("String is :"+Integer.*toString*(number));

}

}

Output:



**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)).

Program:

package org.solution;

public class ques4e {

public static void main(String[] args) {

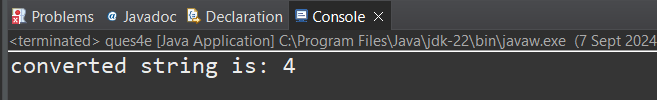
String strnumber ="4";

System.***out***.println("converted string is: "+Integer.*parseInt*(strnumber));

}

}

Output:



**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).

Program:

package org.solution;

public class ques4f {

public static void main(String[] args) {

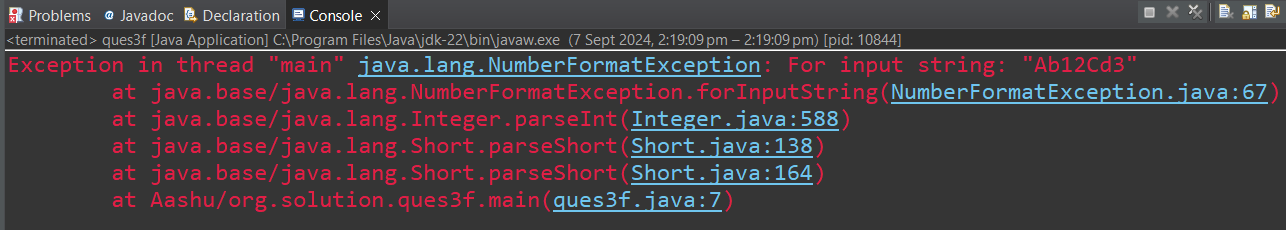
String strNumber ="Ab12Cd3";

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

}

}

Output:



**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)).

Program:

package org.solution;

public class ques4g {

public static void main(String[] args) {

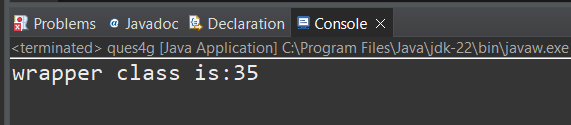
String strnumber = "Ab12Cd3";

int number=35;

System.***out***.println("wrapper class is:"+Integer.*valueOf*(number));

}

}



**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)).

package org.solution;

public class ques4h {

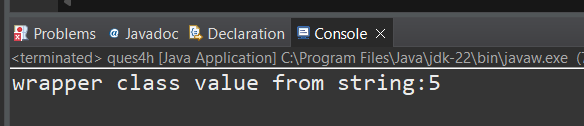
public static void main(String[] args) {

String strnumber = "5";

System.***out***.println("wrapper class value from string:"+Integer.*valueOf*(strnumber));

}

}



**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)).

package org.solution;

public class ques4i {

public static void main(String[] args) {

String strnumber = "5";

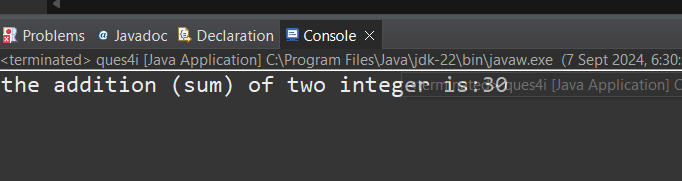
int number =10;

int number2 =20;

System.***out***.println( "the addition (sum) of two integer is:"+Integer.*sum*(number2, number));

}

}



**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)).

package org.solution;

public class ques4j {

public static void main(String[] args) {

String strnumber = "5";

int number =10;

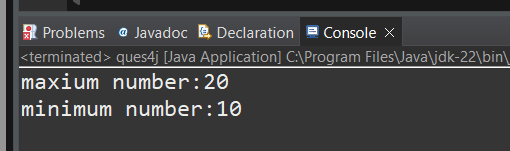
int number2 =20;

System.***out***.println( "maxium number:"+Integer.*max*(number,number2));

System.***out***.println( "minimum number:"+Integer.*min*(number,number2));

}

}



**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)).

package org.solution;

public class ques4k {

public static void main(String[] args) {

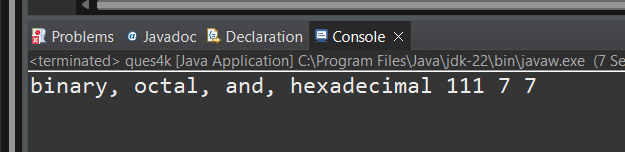
String strnumber = "5";

int number = 7;

System.***out***.println("binary, octal, and, hexadecimal "+Integer.*toBinaryString*(number) +" "+ Integer.*toOctalString*(number) +" "+Integer.*toHexString*(number));

}

}



**l.** Experiment with converting an int value into other primitive types or vice versa and observe the results.

package org.solution;

public class ques4l {

public static void main(String[] args) {

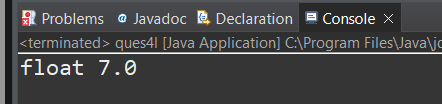
String strnumber = "5";

int number = 7;

System.***out***.println("float "+(float)number);

}

}



**5. Working with java.lang.Long**

**a.** Explore the [Java API documentation for java.lang.Long](https://docs.oracle.com/javase/8/docs/api/java/lang/Long.html) and observe its modifiers and super types.

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

package org.solution;

public class ques5 {

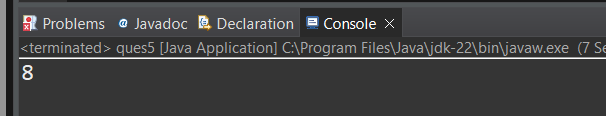
public static void main(String[] args) {

long number = 739781346;

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

}

}



**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).

package org.solution;

public class ques5 {

public static void main(String[] args) {

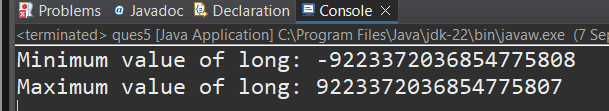
long number = 739781346;

System.***out***.println("Minimum value of long: " + Long.***MIN\_VALUE***);

System.***out***.println("Maximum value of long: " + Long.***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)).

package org.solution;

public class ques5 {

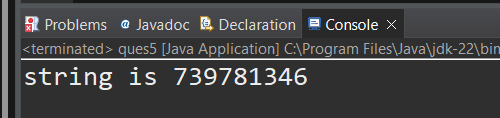
public static void main(String[] args) {

long number = 739781346;

System.***out***.println("string is " + Long.*toString*(number));

}

}



**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)).

package org.solution;

public class ques5 {

public static void main(String[] args) {

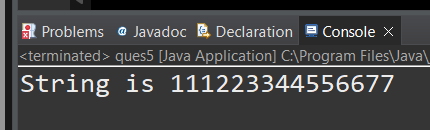
String strNumber ="111223344556677";

long number = 739781346;

System.***out***.println("String is " + Long.*parseLong*(strNumber));

}

}



**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).

package org.solution;

public class ques5 {

public static void main(String[] args) {

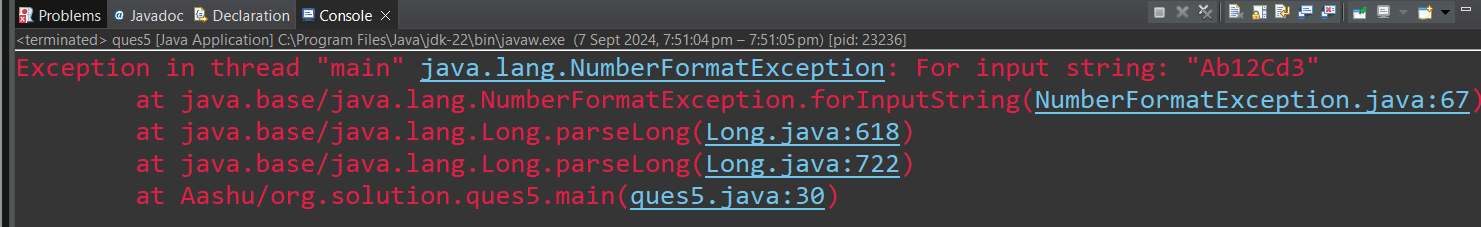
String strnumber="Ab12Cd3";

long number = 739781346;

System.***out***.println("String is " + Long.*parseLong*(strnumber));

}

}



**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)).

package org.solution;

public class ques5 {

public static void main(String[] args) {

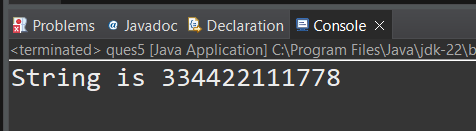
String strnumber="334422111778";

long number = 739781346;

System.***out***.println("String is " + Long.*valueOf*(strnumber));

}

}



1. 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)).

package org.solution;

public class ques5 {

public static void main(String[] args) {

String strnumber="334422111778";

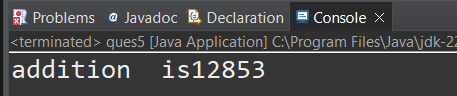
long number1 = 7397;

long number2 = 5456;

System.***out***.println("addition is" + Long.*sum*(number1,number2));

}

}



**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)).

package org.solution;

public class ques5 {

public static void main(String[] args) {

String strnumber="334422111778";

long number1 = 1397;

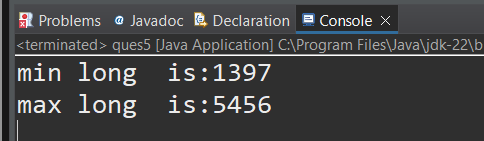
long number2 = 5456;

System.***out***.println("min long is:" + Long.*min*(number1,number2));

System.***out***.println("max long is:" + Long.*max*(number1,number2));

}

}



**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)).

package org.solution;

public class ques5 {

public static void main(String[] args) {

long number = 7;

String binaryString = Long.*toBinaryString*(number);

String octalString = Long.*toOctalString*(number);

String hexString = Long.*toHexString*(number);

System.***out***.println("Binary representation: " + binaryString);

System.***out***.println("Octal representation: " + octalString);

System.***out***.println("Hexadecimal representation: " + hexString);

}

}

**l.** Experiment with converting a long value into other primitive types or vice versa and observe the results.

package org.solution;

public class ques5 {

public static void main(String[] args) {

long longValue = 100L

int intValue = (int) longValue;

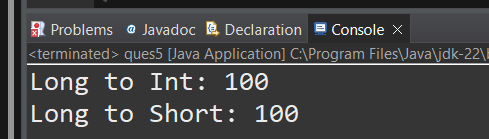
short shortValue = (short) longValue;

System.***out***.println("Long to Int: " + intValue);

System.***out***.println("Long to Short: " + shortValue);

}

}



**6. Working with java.lang.Float**

**a.** Explore the [Java API documentation for java.lang.Float](https://docs.oracle.com/javase/8/docs/api/java/lang/Float.html) and observe its modifiers and super types.

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

package org.solution;

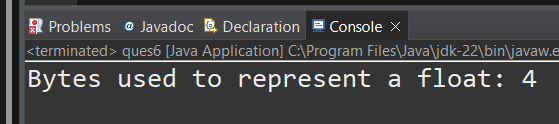
public class ques6 {

public static void main(String[] args) {

System.***out***.println("Bytes used to represent a float: " + Float.***BYTES***);

}

}



**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).

package org.solution;

public class ques6 {

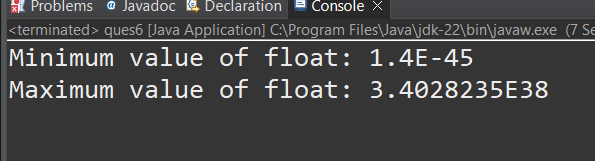
public static void main(String[] args) {

System.***out***.println("Minimum value of float: " + Float.***MIN\_VALUE***);

System.***out***.println("Maximum value of float: " + Float.***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)).

package org.solution;

public class ques6 {

public static void main(String[] args) {

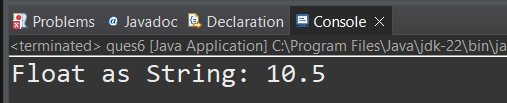
float number = 10.5f;

String floatAsString = Float.*toString*(number);

System.***out***.println("Float as String: " + floatAsString);

}

}



**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)).

package org.solution;

public class ques6 {

public static void main(String[] args) {

System.***out***.println("Hello from ques6!");

}

}

**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).

package org.solution;

public class ques6 {

public static void main(String[] args) {

String strNumber = "11.99";

String strNumber1 = "Ab12Cd3";

float number1 = 11.22f;

float number2 = 55.66f;

int number3 = 20;

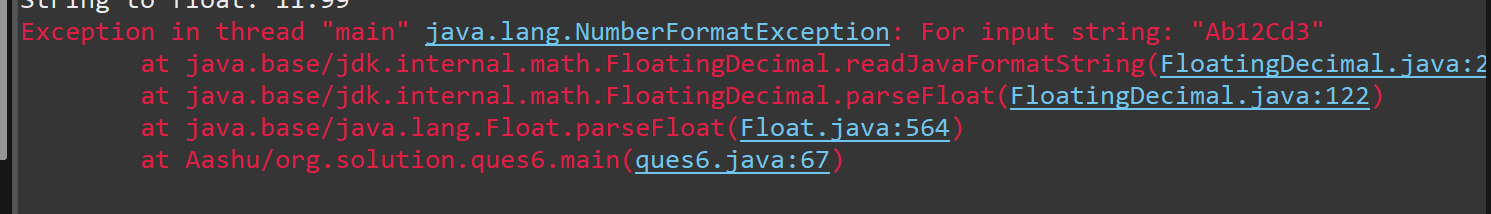
System.***out***.println("Float converted to String: " + Float.*toString*(number1));

System.***out***.println("String to float: " + Float.*parseFloat*(strNumber));

System.***out***.println("String to float: " + Float.*parseFloat*(strNumber1));

}

}



**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)).

package org.solution;

public class ques6 {

public static void main(String[] args) {

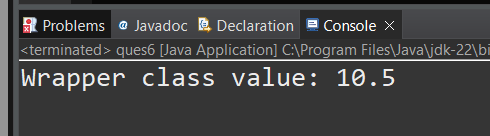
float number = 10.5f;

Float floatWrapper = Float.*valueOf*(number);

System.***out***.println("Wrapper class value: " + floatWrapper);

}

}



**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)).

package org.solution;

public class ques6 {

public static void main(String[] args) {

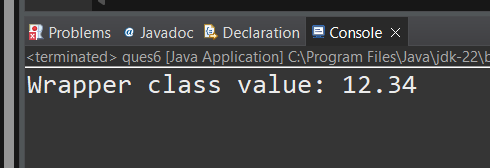
String strNumber = "12.34";

Float floatWrapper = Float.*valueOf*(strNumber)

System.***out***.println("Wrapper class value: " + floatWrapper);

}

}



**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)).

package org.solution;

public class ques6 {

public static void main(String[] args) {

float num1 = 112.3f;

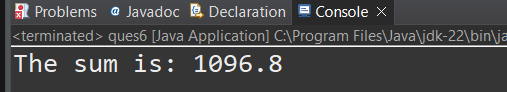
float num2 = 984.5f;

float sum = Float.*sum*(num1, num2);

System.***out***.println("The sum is: " + 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)).

package org.solution;

public class ques6 ;

public static void main(String[] args) {

float num1 = 112.2f;

float num2 = 556.6f;

float minValue = Float.*min*(num1, num2);

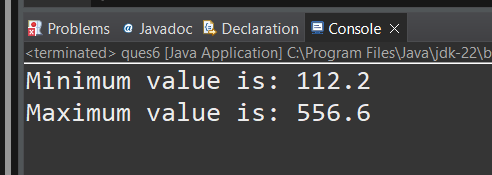
float maxValue = Float.*max*(num1, num2);

System.***out***.println("Minimum value is: " + minValue);

System.***out***.println("Maximum value is: " + maxValue);

}

}



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

package org.solution;

public class ques6 {

public static void main(String[] args) {

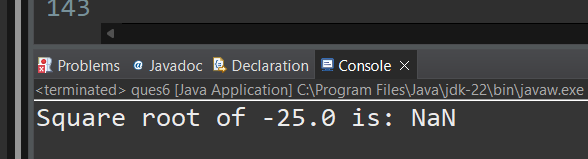
float num = -25.0f;

double sqrtValue = Math.*sqrt*(num);

System.***out***.println("Square root of " + num + " is: " + sqrtValue);

}

}



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

package org.solution;

public class ques6 {

public static void main(String[] args) {

float a = 0.0f;

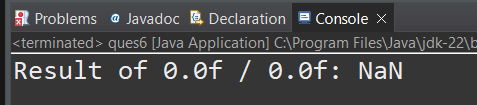
float b = 0.0f;

float result = a / b;

System.***out***.println("Result of 0.0f / 0.0f: " + result);

}

}



**m.** Experiment with converting a float value into other primitive types or vice versa and observe the results.

package org.solution;

public class ques6 {

public static void main(String[] args) {

float floatValue = 10.75f;

int intValue = (int) floatValue;

double doubleValue = floatValue;

float floatFromInt = (float) 5;

float floatFromDouble = (float) 20.99;

System.***out***.println("Original float value: " + floatValue);

System.***out***.println("Float to int: " + intValue);

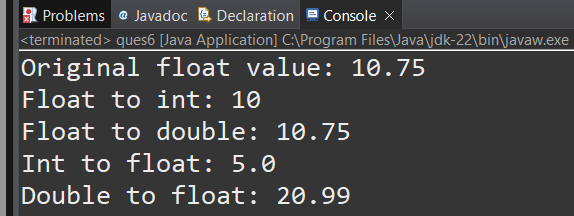
System.***out***.println("Float to double: " + doubleValue);

System.***out***.println("Int to float: " + floatFromInt);

System.***out***.println("Double to float: " + floatFromDouble);

}

}



**7. Working with java.lang.Double**

**a.** Explore the [Java API documentation for java.lang.Double](https://docs.oracle.com/javase/8/docs/api/java/lang/Double.html) and observe its modifiers and super types.

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

package org.solution;

public class ques6 {

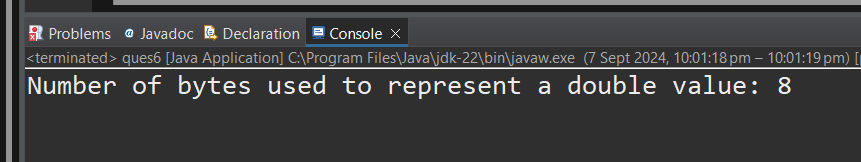
public static void main(String[] args) {

int bytesForDouble = Double.***BYTES***;

System.***out***.println("Number of bytes used to represent a double value: " + bytesForDouble);

}

}



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).

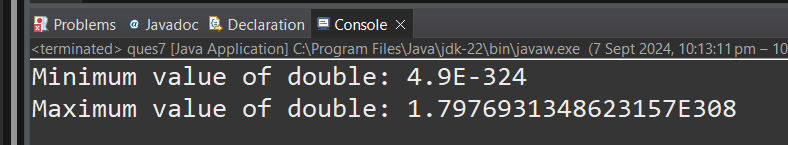
package org.solution;

public class ques7 {

public static void main(String[] args) {

double minValue = Double.MIN\_VALUE;

double maxValue = Double.MAX\_VALUE; System.out.println("Minimum value of double: " + minValue); System.out.println("Maximum value of double: " + maxValue); } }



**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)).

package org.solution;

public class ques7 {

public static void main(String[] args) {

double number = 123.456;

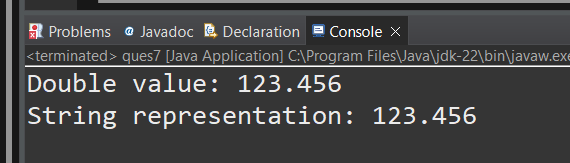
String numberString = Double.*toString*(number);

System.***out***.println("Double value: " + number);

System.***out***.println("String representation: " + numberString);

}

}



**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)).

package org.solution;

public class ques7 {

public static void main(String[] args) {

String strNumber = "123.456"; // Declare and initialize the String variable

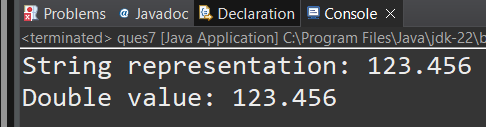
double number = Double.*parseDouble*(strNumber); // Convert String to double

System.***out***.println("String representation: " + strNumber);

System.***out***.println("Double value: " + number);

}

}



**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).

package org.solution;

public class ques7 {

public static void main(String[] args) {

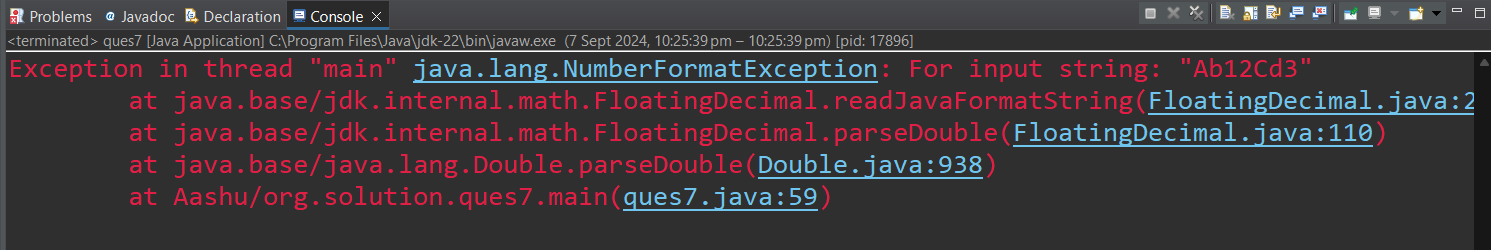
String strNumber = "Ab12Cd3";

double number = Double.*parseDouble*(strNumber);

System.***out***.println("Double value: " + number);

}

}



**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)).

package org.solution;

public class ques7 {

public static void main(String[] args) {

double number = 123.456;

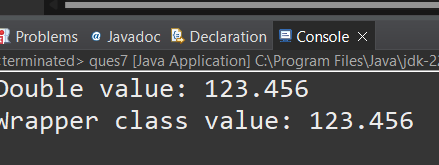
Double wrapperNumber = Double.*valueOf*(number);

System.***out***.println("Double value: " + number);

System.***out***.println("Wrapper class value: " + wrapperNumber);

}

}



**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)).

package org.solution;

public class ques7 {

public static void main(String[] args) {

String strNumber = "123.456";

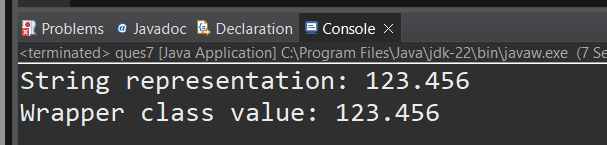
Double wrapperNumber = Double.*valueOf*(strNumber);

System.***out***.println("String representation: " + strNumber);

System.***out***.println("Wrapper class value: " + wrapperNumber);

}

}



1. 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)).

package org.solution;

public class ques7 {

public static void main(String[] args) {

double num1 = 112.3;

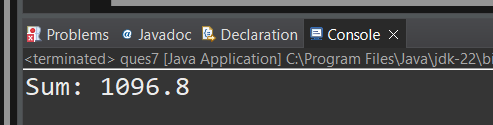
double num2 = 984.5;

double sum = Double.*sum*(num1, num2);

System.***out***.println("Sum: " + 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)).

package org.solution;

public class ques7 {

public static void main(String[] args) {

double num1 = 112.2;

double num2 = 556.6;

double min = Double.*min*(num1, num2);

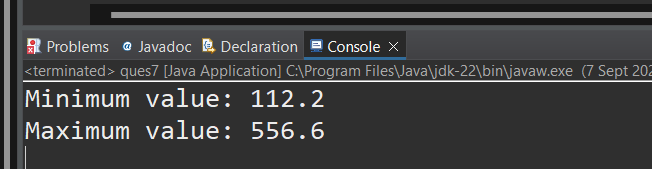
double max = Double.*max*(num1, num2);

System.***out***.println("Minimum value: " + min);

System.***out***.println("Maximum value: " + max);

}

}



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

package org.solution;

public class ques7 {

public static void main(String[] args) {

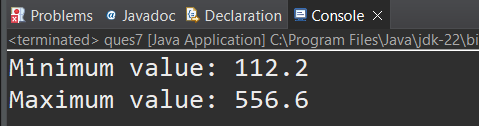
double number = -25.0;

double sqrt = Math.*sqrt*(number);

System.***out***.println("Square root: " + sqrt);

}

}



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

package org.solution;

public class ques7 {

public static void main(String[] args) {

double num1 = 0.0;

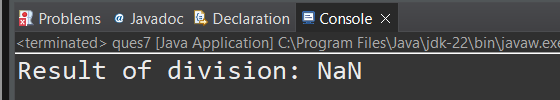
double num2 = 0.0;

double result = num1 / num2;

System.***out***.println("Result of division: " + result);

}

}



**m.** Experiment with converting a double value into other primitive types or vice versa and observe the results.

package org.solution;

public class ques7 {

public static void main(String[] args) {

double number = 123.456;

int intValue = (int) number;

float floatValue = (float) number;

long longValue = (long) number;

byte byteValue = (byte) number;

System.***out***.println("Original double value: " + number);

System.***out***.println("Converted to int: " + intValue);

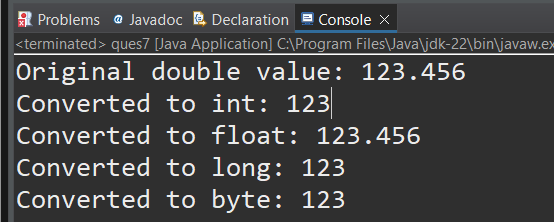
System.***out***.println("Converted to float: " + floatValue);

System.***out***.println("Converted to long: " + longValue);

System.***out***.println("Converted to byte: " + byteValue);

}

}



**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()).

package org.solution;

public class ques8 {

public static void main(String[] args) {

int intValue = 100;

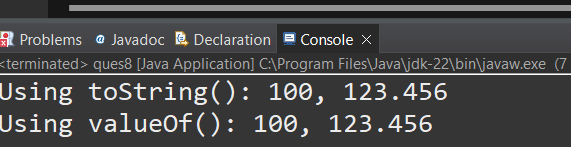
double doubleValue = 123.456;

float floatValue = 45.67f;

long longValue = 100000L;

boolean booleanValue = true;

char charValue = 'A'; System.out.println("Using toString(): " + Integer.toString(intValue) + ", " + Double.toString(doubleValue)); System.out.println("Using valueOf(): " + String.valueOf(intValue) + ", " + String.valueOf(doubleValue)); } }



**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).

package org.solution;

public class ques8 {

int intValue;

double doubleValue;

static int *staticIntValue*;

public static void main(String[] args) {

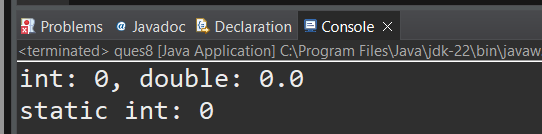
ques8 obj = new ques8();

System.***out***.println("int: " + obj.intValue + ", double: " + obj.doubleValue);

System.***out***.println("static int: " + *staticIntValue*);

}

}



**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).

package org.solution;

public class ques8 {

public static void main(String[] args) {

if (args.length != 3) {

System.***out***.println("Usage: java ArithmeticOperations <num1> <num2> <operator>");

return;

}

try {

int num1 = Integer.*parseInt*(args[0]);

int num2 = Integer.*parseInt*(args[1]);

char operator = args[2].charAt(0);

switch (operator) {

case '+': System.***out***.println(num1 + num2); break;

case '-': System.***out***.println(num1 - num2); break;

case '\*': System.***out***.println(num1 \* num2); break;

case '/':

if (num2 == 0) System.***out***.println("Division by zero");

else System.***out***.println(num1 / num2);

break;

default: System.***out***.println("Invalid operator");

}

} catch (NumberFormatException e) {

System.***out***.println("Please provide valid integers.");

}

}

}