#### **1. Working with** java.lang.Boolean

**a.** 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.

**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) ).

public class b1jbl{

    public static void main(String[] args) {

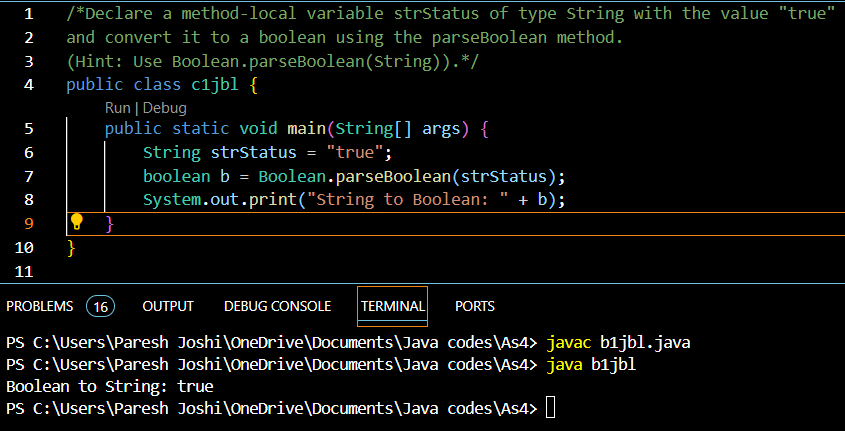
         boolean b = true;

         String bts = Boolean.toString(b);

         System.out.println("Boolean to String: " + bts);

     }

 }



**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 c1jbl {

    public static void main(String[] args) {

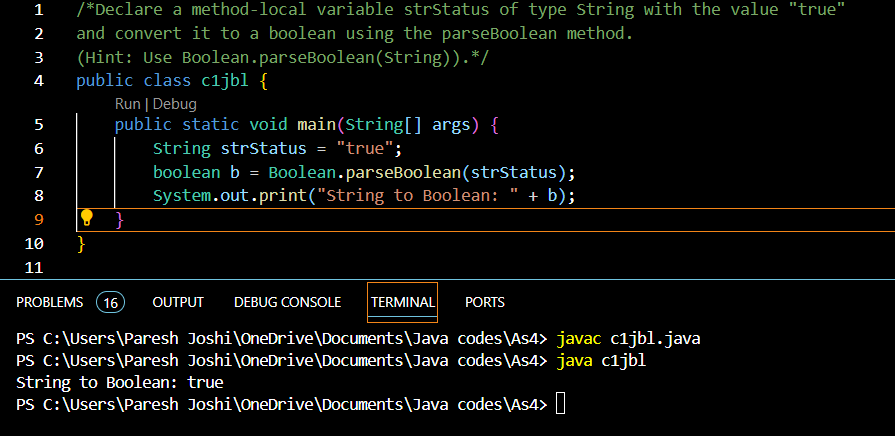
        String strStatus = "true";

        boolean b = Boolean.parseBoolean(strStatus);

        System.out.print("String to Boolean: " + 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 d1jbl {

    public static void main(String[] args) {

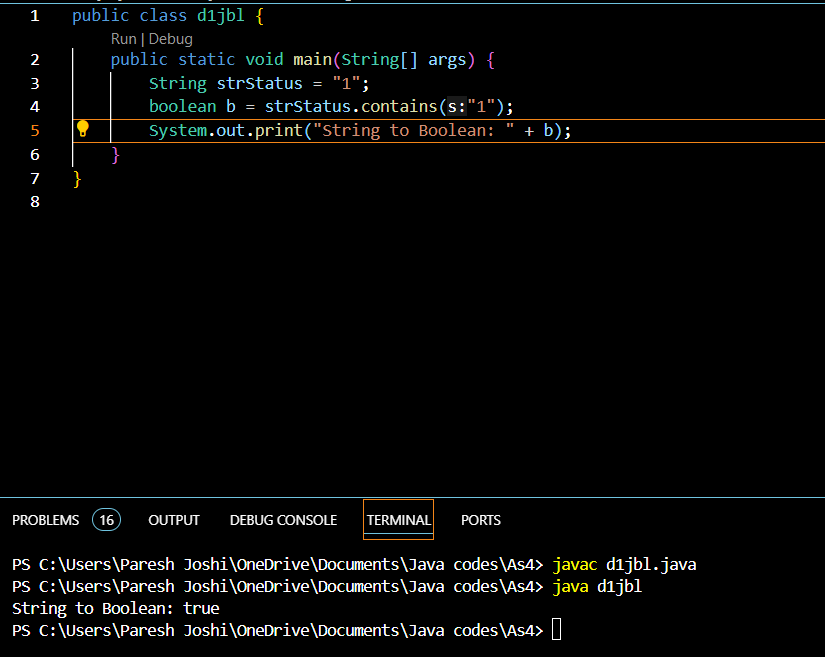
        String strStatus = "1";

        boolean b = strStatus.contains("1");

        System.out.print("String to Boolean: " + b);

    }

}



**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 e1jbl {

    public static void main(String[] args) {

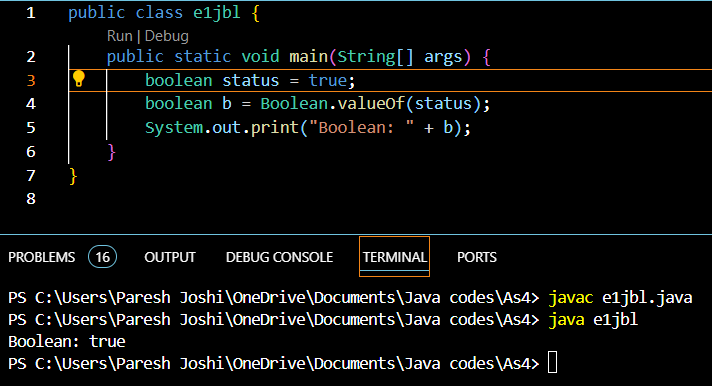
        boolean status = true;

        boolean b = Boolean.valueOf(status);

        System.out.print("Boolean: " + 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 f1jbl {

    public static void main(String[] args) {

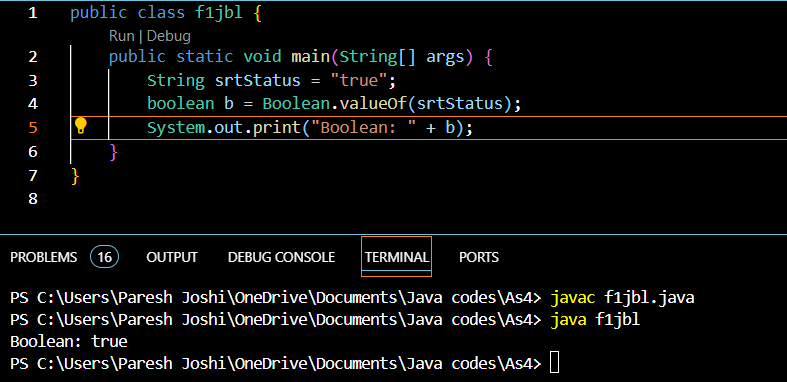
        String srtStatus = "true";

        boolean b = Boolean.valueOf(srtStatus);

        System.out.print("Boolean: " + b);

    }

}



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

public class g1jbl {

    public static void main(String[] args) {

        String str = "1";

        boolean b1 = Boolean.valueOf(str);

        System.out.print("Boolean1: " + b1);

        //Giving false value on any value in string, except true/false.

        /\*incompatible types:

        int num = 1;

        boolean b2 = Boolean.parseBoolean(num);

        System.out.print("Boolean2: " + b2);

        \*/

        /\*incompatible types:

        boolean b3= true;

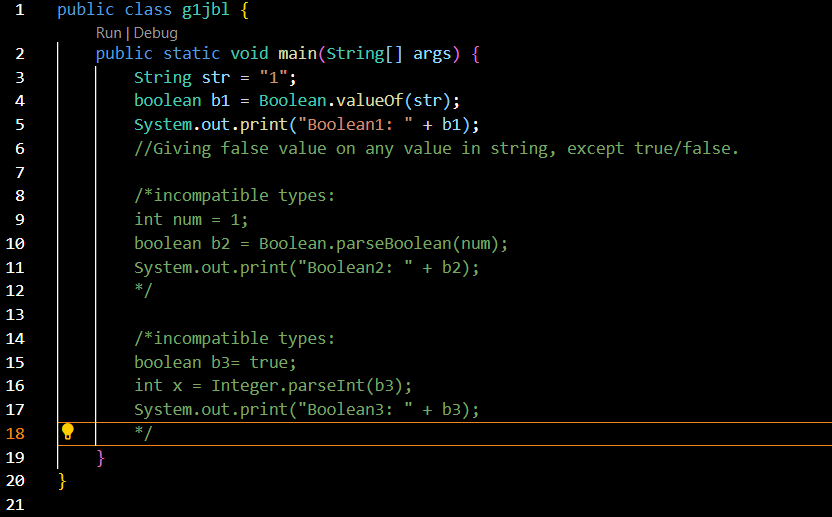
        int x = Integer.parseInt(b3);

        System.out.print("Boolean3: " + b3);

        \*/

    }

}



#### **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.

**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 b2jbl {

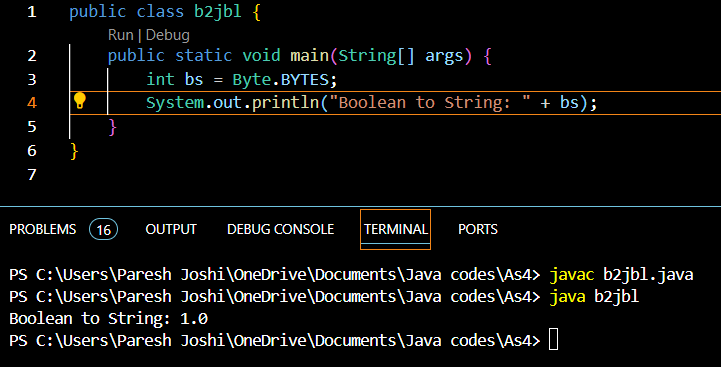
    public static void main(String[] args) {

        int bs = Byte.BYTES;

        System.out.println("Boolean to String: " + bs);

    }

}



**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 c2jbl {

    public static void main(String[] args) {

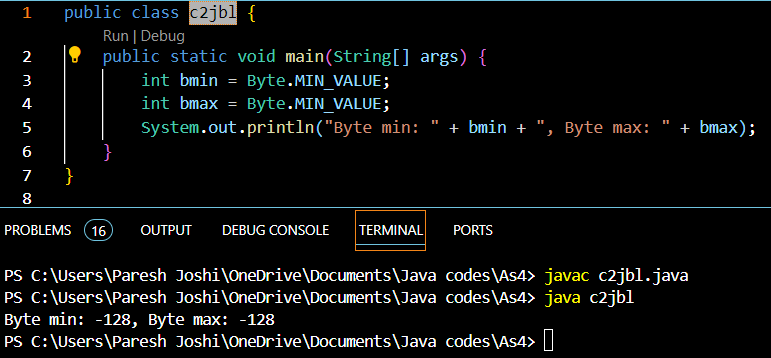
        int bmin = Byte.MIN\_VALUE;

        int bmax = Byte.MIN\_VALUE;

        System.out.println("Byte min: " + bmin + ", Byte max: " + bmax);

    }

}



**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 d2jbl {

    public static void main(String[] args) {

        byte num = 65;

        String byts = Byte.toString(num);

        System.out.println("Byte min: " + num + ", Byte max: " + byts);

    }

}



**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 e2jbl {

    public static void main(String[] args) {

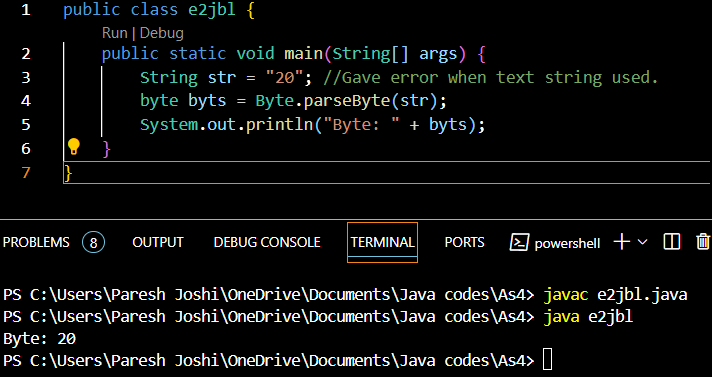
        String str = "20"; //Gave error when text string used.

        byte byts = Byte.parseByte(str);

        System.out.println("Byte: " + byts);

    }

}



**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 f2jbl {

    public static void main(String[] args) {

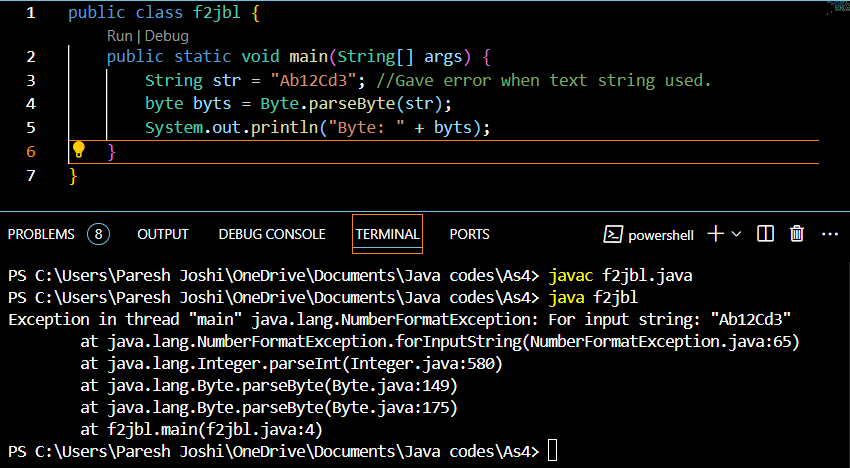
        String str = "Ab12Cd3"; //Gave error when text string used.

        byte byts = Byte.parseByte(str);

        System.out.println("Byte: " + byts);

    }

}



**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 h2jbl {**

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

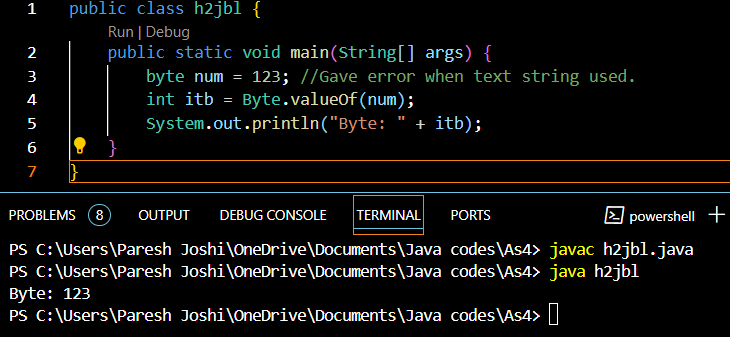
**byte num = 123; //Gave error when text string used.**

**int itb = Byte.valueOf(num);**

**System.out.println("Byte: " + itb);**

**}**

**}**

****

**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 g2jbl {

    public static void main(String[] args) {

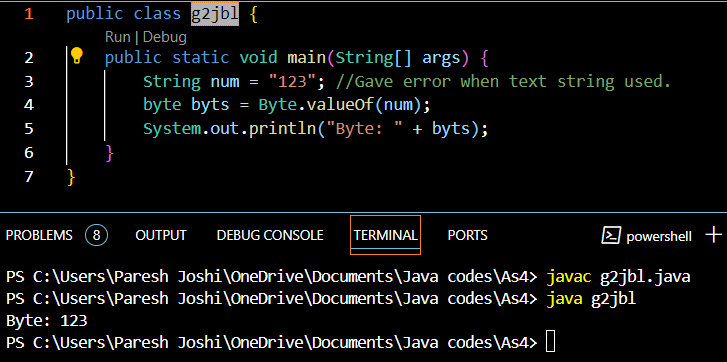
        String num = "123"; //Gave error when text string used.

        byte byts = Byte.valueOf(num);

        System.out.println("Byte: " + byts);

    }

}



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

#### **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).

**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).

**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)).

**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)).

**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).

**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)).

**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)).

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

public class jbl3 {

    public static void main(String[] args) {

        String strNumber ="12651";

        String strNumber1 ="Ab12Cd3";

        short number1=11102;

        short number2=9846;

        int number3=156891;

        System.out.println("Bytes used to represent short: " + Short.BYTES); //3B

        System.out.println("Min value of short: " + Short.MIN\_VALUE + " Min value of short: "+ Short.MAX\_VALUE); //3C

        System.out.println("Short converted to short:" + Short.toString(number1)); //3D

        System.out.println("String to short:    " + Short.parseShort(strNumber)); //3E

        System.out.println("Short to wrapper class: " + Short.valueOf(number2)); //3G

        System.out.println("String to wrapper class:    " + Short.valueOf(strNumber)); //3H

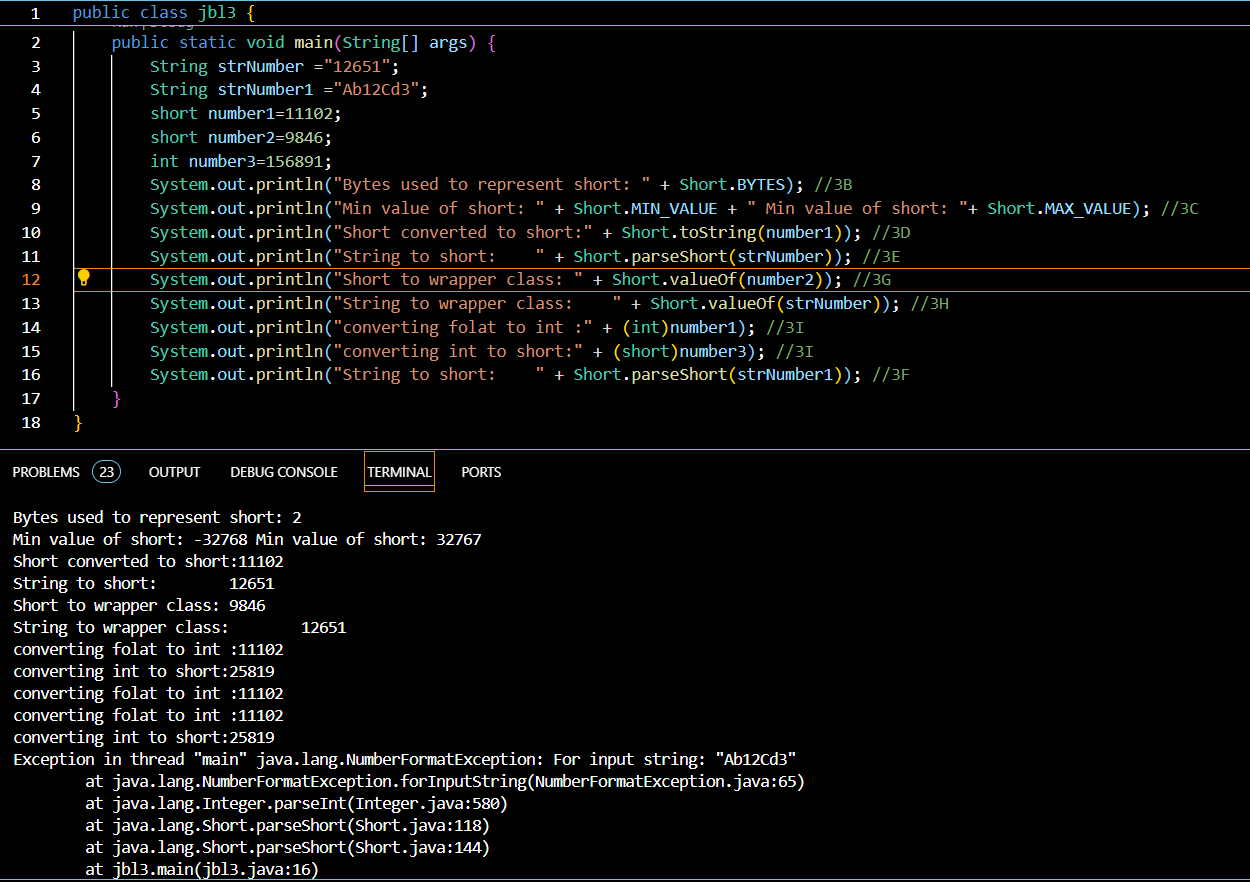
        System.out.println("converting folat to int :" + (int)number1); //3I

        System.out.println("converting int to short:" + (short)number3); //3I

        System.out.println("String to short:    " + Short.parseShort(strNumber1)); //3F

    }

}



#### **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).

public class b3jbl {

    public static void main(String[] args) {

        int x = Integer.BYTES;

        System.out.println(x);

    }

}

OP - 4

**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 c3jbl {

    public static void main(String[] args) {

        int int\_min = Integer.MIN\_VALUE;

        int int\_max = Integer.MAX\_VALUE;

        System.out.println("Byte min: " + int\_min + ", Byte max: " + int\_max);

    }

}

OP - Byte min: -2147483648, Byte max: 2147483647

**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 d3jbl {

    public static void main(String[] args) {

        int num = 123;

        String str = Integer.toString(num);

        System.out.println(str);

    }

}

OP -123

**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 e3jbl {

    public static void main(String[] args) {

        String str = "123";

        int num = Integer.parseInt(str);

        System.out.println(num);

    }

}

OP - 123

**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 b3jbl {

    public static void main(String[] args) {

        String str = "Ab12Cd3";

        int num = Integer.parseInt(str);

        System.out.println(num);

    }

}

OP - Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"

at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)

at java.lang.Integer.parseInt(Integer.java:580)

at java.lang.Integer.parseInt(Integer.java:615)

at b3jbl.main(b3jbl.java:4)

**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 b3jbl {

    public static void main(String[] args) {

        int num2 = 123;

        int num = Integer.valueOf(num2);

        System.out.println(num);

    }

}

OP - 123

**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 b3jbl {

    public static void main(String[] args) {

        String num2 = "123";

        int num = Integer.valueOf(num2);

        System.out.println(num);

    }

}

OP - 123

**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 b3jbl {

    public static void main(String[] args) {

        int num1 = 10;

        int num2 = 10;

        int sum = Integer.sum(num1, num2);

        System.out.println(sum);

    }

}

OP - 20

**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 b3jbl {

    public static void main(String[] args) {

        int num1 = 10;

        int num2 = 10;

        int max = Integer.max(num1, num2);

        int min = Integer.min(num1, num2);

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

    }

}

OP - Min 10 Min 10

**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 b3jbl {

    public static void main(String[] args) {

        int num1 = 7;

        int bin = Integer.parseInt(Integer.toBinaryString(num1));

        int oct = Integer.parseInt(Integer.toOctalString(num1));;

        int hex = Integer.parseInt(Integer.toHexString(num1));;

        System.out.println("Bin " + bin + " Oct " + oct + " Hex " + hex);

    }

}

OP - Bin 111 Oct 7 Hex 7

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

#### **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).

**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).

**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)).

**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)).

**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).

**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)).

**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)).

**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)).

**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)).

**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)).

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

public class jbl5 {

    public static void main(String[] args) {

        String strNumber ="12651";

        String strNumber1 ="Ab12Cd3";

        long number1=11102;

        long number2=9846;

        int number3=156891;

        long number4=9845;

        System.out.println("Bytes used to represent long: " + Long.BYTES); //5B

        System.out.println("Min value of long: " + Long.MIN\_VALUE + " Min value of long: "+ Long.MAX\_VALUE); //5C

        System.out.println("Long converted to long:" + Long.toString(number1)); //5D

        System.out.println("String to long: " + Long.parseLong(strNumber)); //5E

        System.out.println("Long to wrapper class:  " + Long.valueOf(number2)); //5G

        System.out.println("String to wrapper class:    " + Long.valueOf(strNumber)); //5H

        System.out.println("sum of floats    :  " + Long.sum(number1, number2)); //5I

        System.out.println("Minimum of floats:  " + Long.min(number1, number2) + " Maximum of floats: " + Long.max(number1, number2)); //5J

        System.out.println("Long to Binary:" + (Long.toBinaryString(number4))); //5k

        System.out.println("Long to Headecimal:" + (Long.toHexString(number4))); //5k

        System.out.println("Long to Octal:" + (Long.toOctalString(number4))); //5k

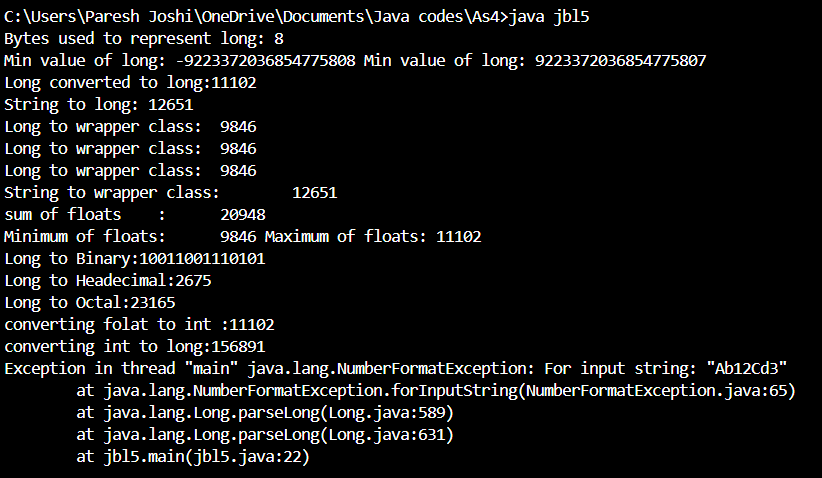
        System.out.println("converting folat to int :" + (int)number1); //5L

        System.out.println("converting int to long:" + (long)number3); //5L

        System.out.println("String to long: " + Long.parseLong(strNumber1)); //5F

    }

}



#### **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).

**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).

**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)).

**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)).

**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).

**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)).

**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)).

**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)).

**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)).

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

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

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

public class jbl6 {

    public static void main(String[] args) {

        String strNumber ="11.99";

        String strNumber1 ="Ab12Cd3";

        float number1=112.3f;

        float number2=984.5f;

        float number4=-25.0f;

        float number6=0.0f;

        float number7=0.0f;

        int number3=20;

        System.out.println("Bytes used to represent long: " + Long.BYTES); //6B

        System.out.println("Min value of long: " + Long.MIN\_VALUE + " Min value of long: "+ Long.MAX\_VALUE); //6C

        System.out.println("Float converted to float:" + Float.toString(number1)); //6D

        System.out.println("String to float:    " + Float.parseFloat(strNumber)); //6E

        System.out.println("float to wrapper class: " + Float.valueOf(number1)); //6G

        System.out.println("float to wrapper class: " + Float.valueOf(strNumber)); //6H

        System.out.println("sum of floats    :  " + Float.sum(number1, number2)); //6I

        System.out.println("Minimum of floats:  " + Float.min(number1, number2)); //6J

        System.out.println("Maximum of floats:  " + Float.max(number1, number2)); //6J

        System.out.println("Square root of float:" + Math.sqrt(number4)); //6K

        System.out.println("Float divide by float:" + (number6/number7)); //6L

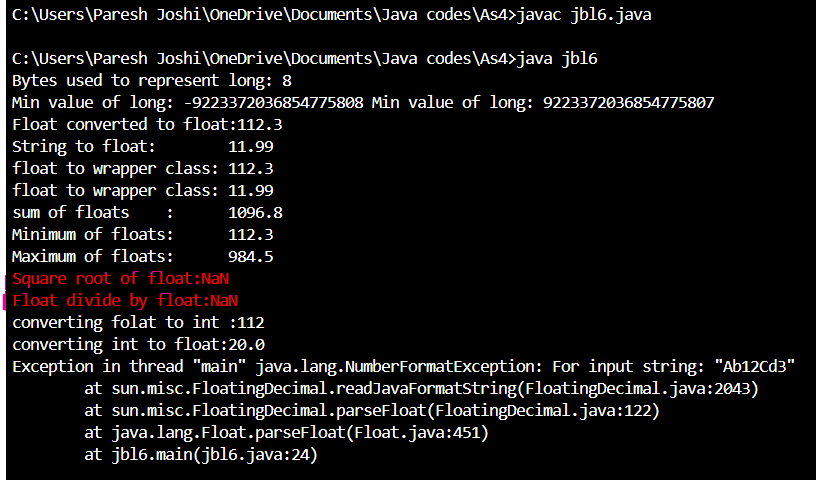
        System.out.println("converting folat to int :" + (int)number1); //6M

        System.out.println("converting int to float:" + (float)number3); //6M

        System.out.println("String to float:    " + Float.parseFloat(strNumber1)); //6F

    }

}



#### **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).

**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).

**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)).

**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)).

**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).

**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)).

**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)).

**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)).

**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)).

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

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

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

public class jbl7 {

    public static void main(String[] args) {

        String strNumber ="11.99";

        String strNumber1 ="Ab12Cd3";

        double number1=112.3f;

        double number2=984.5f;

        double number4=-25.0f;

        double number6=0.0f;

        double number7=0.0f;

        int number3=20;

        System.out.println("Bytes used to represent long: " + Long.BYTES); //7B

        System.out.println("Min value of long: " + Long.MIN\_VALUE + " Min value of long: "+ Long.MAX\_VALUE); //7C

        System.out.println("Double converted to double:" + Double.toString(number1)); //7D

        System.out.println("String to double:   " + Double.parseDouble(strNumber)); //7E

        System.out.println("double to wrapper class:    " + Double.valueOf(number1)); //7G

        System.out.println("double to wrapper class:    " + Double.valueOf(strNumber)); //7H

        System.out.println("sum of doubles    : " + Double.sum(number1, number2)); //7I

        System.out.println("Minimum of doubles: " + Double.min(number1, number2)); //7J

        System.out.println("Maximum of doubles: " + Double.max(number1, number2)); //7J

        System.out.println("Square root of double:" + Math.sqrt(number4)); //7K

        System.out.println("Double divide by double:" + (number6/number7)); //7L

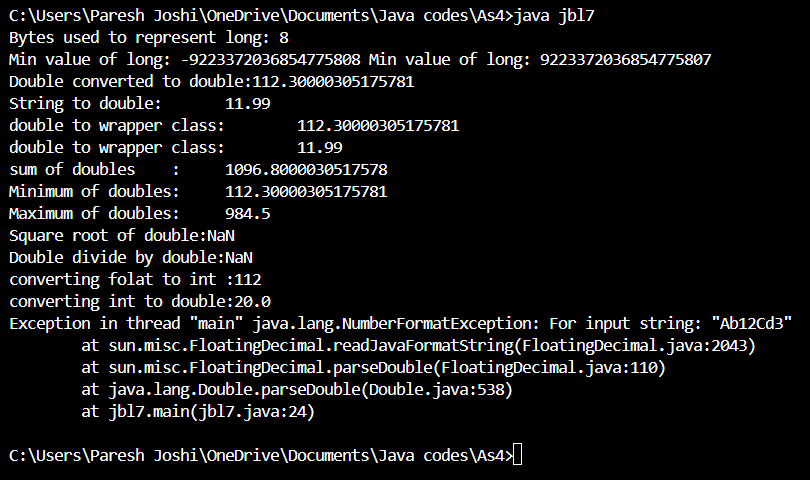
        System.out.println("converting folat to int :" + (int)number1); //7M

        System.out.println("converting int to double:" + (double)number3); //7M

        System.out.println("String to double:   " + Double.parseDouble(strNumber1)); //7F

    }

}



#### **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()).
* import java.util.Scanner;
* public class jbl8 {
* public static void main(String[] args) {
* Scanner scObj = new Scanner(System.in);
* System.out.println("Enter Integer: ");
* int num1 = scObj.nextInt();
* String a1 = Integer.toString(num1);
* String a2 = String.valueOf(num1);
* System.out.println(a1 + " " + a2);
* System.out.println("Enter Float: ");
* float num2 = scObj.nextFloat();
* String b1 = Float.toString(num2);
* String b2 = String.valueOf(num2);
* System.out.println(b1 + " " + b2);
* System.out.println("Enter Short: ");
* short num3 = scObj.nextShort();
* String c1 = Short.toString(num3);
* String c2 = String.valueOf(num3);
* System.out.println(c1 + " " + c2);
* System.out.println("Enter Long: ");
* Long num4 = scObj.nextLong();
* String d1 = Long.toString(num4);
* String d2 = String.valueOf(num4);
* System.out.println(d1 + " " + d2);
* System.out.println("Enter Double: ");
* double num5 = scObj.nextDouble();
* String e1 = Double.toString(num5);
* String e2 = String.valueOf(num5);
* System.out.println(e1 + " " + e2);
* System.out.println("Enter Byte: ");
* byte num6 = scObj.nextByte();
* String f1 = Integer.toString(num6);
* String f2 = String.valueOf(num6);
* System.out.println(f1 + " " + f2);
* System.out.println("Enter Boolean: ");
* boolean num7 = scObj.nextBoolean();
* String g1 = Boolean.toString(num7);
* String g2 = String.valueOf(num7);
* System.out.println(g1 + " " + g2);
* scObj.close();
* }
* }

#### **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).

class Premetive {

    byte by;

    int in;

    float fl;

    char ch;

    short sh;

    long lg;

    double db;

    String str;

    public void printVals() {

        System.out.println("Byte: " + this.by);

        System.out.println("Int: " + this.in);

        System.out.println("Float: " + this.fl);

        System.out.println("Char: " + this.ch);

        System.out.println("Short: " + this.sh);

        System.out.println("Long: " + this.lg);

        System.out.println("Double: " + this.db);

        System.out.println("String: " + this.str);

    }

}

public class A2Q9 {

    public static void main(String[] args) {

        Premetive pr = new Premetive();

        pr.printVals();

    }

}

OP - Byte: 0

Int: 0

Float: 0.0

Char:

Short: 0

Long: 0

Double: 0.0

String: null

#### **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).

public class A2Q10 {

    public static void main(String[] args) {

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

        String opr = args[1];

        int num2 = Integer.parseInt(args[2]);

        switch (opr) {

            case "+":

                System.out.println(num1+num2);

                break;

            case "-":

                System.out.println(num1-num2);

                break;

            case "\*":

                System.out.println(num1\*num2);

                break;

            case "/":

                System.out.println(num1/num2);

                break;

            default:

                System.out.println("Failed");

                break;

        }

    }

}

