# 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 B {**

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

**boolean value = true;**

**String str = Boolean.toString(value);**

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

**}**

**}**

**Output = true**

**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 C {

    public static void main(String[] args) {

        String strStatus = "true";

        boolean value = Boolean.parseBoolean(strStatus);

        System.out.println(value);

    }

}

Output : true

**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 D {

    public static void main(String[] args) {

        String strStatus = "0";

        if (strStatus.equals("0")){

            strStatus = "false";

        } else if (strStatus.equals("1")) {

            strStatus = "true";

        }

        boolean value = Boolean.parseBoolean(strStatus);

        System.out.println(value);

    }

}

Output : false

**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 E {**

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

**boolean val = true;**

**System.out.println(Boolean.valueOf(val));**

**}**

**}**

**Output = true**

**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 F {**

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

**String strStatus = "true";**

**System.out.println(Boolean.valueOf(strStatus));**

**}**

**}**

**Output : true**

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

**Code:**

**public class G {  
 boolean val = true;  
 char ch = (char)val;  
}**

Ans = Error occurred : **inconvertible types; cannot cast 'boolean' to 'char'**

# 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).

**code:**

**public class B {  
 public static void main(String[] args) {  
 System.*out*.println(Byte.*BYTES*);  
 }  
}**

**Output : 1**

**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 C {**

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

**System.out.println("Minimum byte value : "+ Byte.MIN\_VALUE);**

**System.out.println("Maximum byte value : "+ Byte.MAX\_VALUE);**

**}**

**}**

**Output:**

**Minimum byte value : -128**

**Maximum byte value : 127**

**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 D {**

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

**byte num = 123;**

**String str = Byte.toString(num);**

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

**}**

**}**

**Output : 123**

**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 E {**

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

**String strNumber = "111";**

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

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

**}**

**Output : 111**

**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).

Ans:

**public class F {**

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

**String strNumber = "Ab12Cd3";  // Method-local variable**

**byte number = Byte.parseByte(strNumber);  // Attempting to convert to byte**

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

**}**

**}**

**Output:**

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

**at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)**

**at java.base/java.lang.Integer.parseInt(Integer.java:668)**

**at java.base/java.lang.Byte.parseByte(Byte.java:193)**

**at java.base/java.lang.Byte.parseByte(Byte.java:219)**

**at F.main(F.java:4)**

**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 G {**

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

**byte num = 123;**

**System.out.println(Byte.valueOf(num));**

**}**

**}**

**Output : 123**

**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 H {**

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

**String strNum = "123";**

**System.out.println(Byte.valueOf(strNum));**

**}**

**}**

**Output : 123**

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

**public class I {**

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

**byte num = 117;**

**int num2 = num;**

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

**}**

**}**

**Output: 117**

# **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).

**public class B {**

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

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

**}**

**}**

**Output : 2**

**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 C {**

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

**short min = Short.MIN\_VALUE;**

**short max = Short.MAX\_VALUE;**

**System.out.println("Max Value: "+max);**

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

**}**

**}**

**Output:**

**Max Value: 32767**

**min Value: -32768**

**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 D {**

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

**Short num = 1234;**

**String str = Short.toString(num);**

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

**}**

**}**

**Output = 1234**

**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 E {**

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

**String strNumber = "111";**

**Short num = Short.parseShort(strNumber);**

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

**}**

**}**

**Output = 111**

**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 F {**

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

**String strNumber = "Ab12Cd3";  // Method-local variable**

**short number = Short.parseShort(strNumber);  // Attempting to convert to short**

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

**}**

**}**

**Output:**

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

**at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)**

**at java.base/java.lang.Integer.parseInt(Integer.java:668)**

**at java.base/java.lang.Short.parseShort(Short.java:137)**

**at java.base/java.lang.Short.parseShort(Short.java:163)**

**at F.main(F.java:4)**

**PS D:\Cdac\OOP Java\Assignments\Sandeep sir's Assignments\Assignment 2 codes\Q3>**

**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 G {**

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

**Short num = 123;**

**System.out.println(Short.valueOf(num));**

**}**

**}**

**Output: 123**

**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 H {**

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

**String strNum = "123";**

**System.out.println(Short.valueOf(strNum));**

**}**

**}**

**Output: 123**

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

**public class I {**

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

**Short num = 117;**

**int num2 = num;**

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

**}**

**}**

**Output: 117**

# 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 B {**

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

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

**}**

**}**

**Output : 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 C {**

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

**System.out.println("Minimum byte value : "+ Integer.MIN\_VALUE);**

**System.out.println("Maximum byte value : "+ Integer.MAX\_VALUE);**

**}**

**}**

**Output:**

**Minimum byte value : -2147483648**

**Maximum byte value : 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 D {**

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

**int num = 123;**

**String str = Integer.toString(num);**

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

**}**

**}**

**Output: 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 E {**

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

**String strNumber = "111";**

**int num = Integer.parseInt(strNumber);**

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

**}**

**}**

**Output: 111**

**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 F {**

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

**String strNumber = "Ab12Cd3";  // Method-local variable**

**int number = Integer.parseInt(strNumber);  // Attempting to convert to byte**

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

**}**

**}**

**Output:**

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

**at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)**

**at java.base/java.lang.Integer.parseInt(Integer.java:668)**

**at java.base/java.lang.Integer.parseInt(Integer.java:786)**

**at F.main(F.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 G {**

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

**int num = 123;**

**System.out.println(Integer.valueOf(num));**

**}**

**}**

**Output : 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 H {**

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

**String strNum = "123";**

**System.out.println(Integer.valueOf(strNum));**

**}**

**}**

**Output : 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 I {**

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

**int num1 = 19;**

**int num2 = 11;**

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

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

**}**

**}**

**Output : 30**

**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 J {**

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

**int num1 = 19;**

**int num2 = 11;**

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

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

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

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

**}**

**}**

**Output :**

**Minimum value : 11**

**Maximum value : 19**

**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 K {**

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

**int n = 7;**

**System.out.println("Binary String : "+Integer.toBinaryString(n));**

**System.out.println("Octal String : "+Integer.toOctalString(n));**

**System.out.println("Hexadecimal String : "+Integer.toHexString(n));**

**}**

**}**

**Output:**

**Binary String : 111**

**Octal String : 7**

**Hexadecimal String : 7**

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

**public class L {**

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

**int num = 500000;**

**short num2 = (short) num;**

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

**}**

**}**

**Output : -24288**

# 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.

Ans : Read only

**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 B {**

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

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

**}**

**}**

**Output: 4**

**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 C {**

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

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

**System.out.println("Maximum byte value : "+ Long.MAX\_VALUE);**

**}**

**}**

**Output:**

**Minimum byte value : -9223372036854775808**

**Maximum byte value : 9223372036854775807**

**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 D {**

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

**long num = 123;**

**String str = Long.toString(num);**

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

**}**

**}**

**Output: 123**

**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 E {**

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

**String strNumber = "111";**

**long num = Long.parseLong(strNumber);**

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

**}**

**}**

**Output: 111**

**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 F {**

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

**String strNumber = "Ab12Cd3";  // Method-local variable**

**long number = Long.parseLong(strNumber);  // Attempting to convert to byte**

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

**}**

**}**

**Output:**

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

**at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)**

**at java.base/java.lang.Long.parseLong(Long.java:711)**

**at java.base/java.lang.Long.parseLong(Long.java:836)**

**at F.main(F.java:4)**

**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 G {**

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

**long num = 12345678;**

**System.out.println(Long.valueOf(num));**

**}**

**}**

**Output : 12345678**

**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 H {**

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

**String strNum = "123";**

**System.out.println(Long.valueOf(strNum));**

**}**

**}**

**Output: 123**

**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 I {**

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

**long num1 = 1123;**

**long num2 = 9845;**

**long sum = Long.sum(num1, num2);**

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

**}**

**}**

**Output: 10968**

**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 J {**

**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("Minimum value : "+ min);**

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

**}**

**}**

**Output:**

**Minimum value : 1122**

**Maximum value : 5566**

**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 K {**

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

**long n = 7;**

**System.out.println("Binary String : "+Long.toBinaryString(n));**

**System.out.println("Octal String : "+Long.toOctalString(n));**

**System.out.println("Hexadecimal String : "+Long.toHexString(n));**

**}**

**}**

**Output :**

**Binary String : 111**

**Octal String : 7**

**Hexadecimal String : 7**

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

**public class L {**

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

**long num = 500000;**

**short num2 = (short) num;**

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

**}**

**}**

**Output:**

**-24288**

# 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.

Ans : Read only .

**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 B {**

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

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

**}**

**}**

**Output : 4**

**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).

**public class C {**

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

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

**System.out.println("Maximum byte value : "+ Float.MAX\_VALUE);**

**}**

**}**

**Output:**

**Minimum byte value : 1.4E-45**

**Maximum byte value : 3.4028235E38**

**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)).

**public class D {**

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

**Float num = 123.50f;**

**String str = Float.toString(num);**

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

**}**

**}**

**Output: 123.5**

**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)).

**public class E {**

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

**String strNumber = "111.11";**

**Float num = Float.parseFloat(strNumber);**

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

**}**

**}**

**Output: 111.11**

**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).

**public class F {**

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

**String strNumber = "Ab12Cd3";  // Method-local variable**

**float number = Float.parseFloat(strNumber);  // Attempting to convert to byte**

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

**}**

**}**

**Output:**

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

**at java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)**

**at java.base/jdk.internal.math.FloatingDecimal.parseFloat(FloatingDecimal.java:122)**

**at java.base/java.lang.Float.parseFloat(Float.java:476)**

**at F.main(F.java:4)**

**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 G {**

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

**Float num = 12345.678f;**

**System.out.println(Float.valueOf(num));**

**}**

**}**

**Output: 12345.678**

**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 H {**

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

**String strNum = "123.5";**

**System.out.println(Float.valueOf(strNum));**

**}**

**}**

**Output: 123.5**

**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 I {**

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

**float num1 = 112.3f;**

**float num2 = 984.5f;**

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

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

**}**

**}**

**Output:1096.8**

**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 J {**

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

**float num1 = 112.2f;**

**float num2 = 55.66f;**

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

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

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

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

**}**

**}**

**Output:**

**Minimum value : 55.66**

**Maximum value : 112.2**

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

**public class K {**

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

**Float n = -25.0f;**

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

**}**

**}**

**Output: NaN**

**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 L {

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

**Float num = 0.0f;**

**Float num2 = 0.0f;**

**System.out.println(num/num2);**

**}**

**}**

**Output: NaN**

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

**public class M {**

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

**float num = 50.94f;**

**int num2 = (int) num;**

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

**}**

**}**

**Output: 50**

# 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).

Code:

**public class B {**

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

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

**}**

**}**

Output: **8**

**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).

Code:

**public class C {**

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

**System.out.println("Minimum byte value : "+ Double.MIN\_VALUE);**

**System.out.println("Maximum byte value : "+ Double.MAX\_VALUE);**

**}**

**}**

**Output:**

**Minimum byte value : 4.9E-324**

**Maximum byte value : 1.7976931348623157E308**

**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 D {**

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

**double num = 123456;**

**String str = Double.toString(num);**

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

**}**

**}**

**Output:**

**123456.0**

**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 E {**

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

**String strNumber = "111.23";**

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

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

**}**

**}**

**Output: 111.23**

**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 F {**

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

**String strNumber = "Ab12Cd3";  // Method-local variable**

**double number = Double.parseDouble(strNumber);  // Attempting to convert to byte**

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

**}**

}

**Output : Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"java.base/jdk.internal.math.FloatingDecimal.readJavaFormatString(FloatingDecimal.java:2054)**

**at java.base/jdk.internal.math.FloatingDecimal.parseDouble(FloatingDecimal.java:110)**

**at java.base/java.lang.Double.parseDouble(Double.java:651)**

**at F.main(F.java:4)**

**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 G {**

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

**double num = 123.45678;**

**System.out.println(Double.valueOf(num));**

**}**

**}**

**Output : 123.45678**

**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 H {**

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

**String strNum = "123";**

**System.out.println(Double.valueOf(strNum));**

**}**

**}**

**Output : 123.0**

**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 I {**

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

**double num1 = 112.3;**

**double num2 = 984.5;**

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

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

**}**

**}**

**Output : 1096.8**

**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 J {**

**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);**

**}**

**}**

**Output:**

**Minimum value : 112.2**

**Maximum value : 556.6**

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

**public class K {**

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

**double num = -25.0;**

**System.out.println(Math.sqrt(num));**

**}**

**}**

**Output : NaN**

**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 L {**

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

**double num = 0.0d;**

**double num2 = 0.0d;**

**System.out.println(num/num2);**

**}**

**}**

**Output : NaN**

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

**public class M {**

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

**double num = 30.50d;**

**int num2 = (int)num;**

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

**}**

**}**

**Output : 30**

# 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 PrimitiveToString {**

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

**// Initialize variables of all primitive types**

**byte byteValue = 10;**

**short shortValue = 20;**

**int intValue = 42;**

**long longValue = 100000L;**

**float floatValue = 3.14f;**

**double doubleValue = 3.14159;**

**boolean boolValue = true;**

**char charValue = 'A';**

**// Print using the toString method of wrapper classes**

**System.out.println("Using toString method:");**

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

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

**System.out.println("Integer: " + Integer.toString(intValue));**

**System.out.println("Long: " + Long.toString(longValue));**

**System.out.println("Float: " + Float.toString(floatValue));**

**System.out.println("Double: " + Double.toString(doubleValue));**

**System.out.println("Boolean: " + Boolean.toString(boolValue));**

**System.out.println("Character: " + Character.toString(charValue));**

**// Print using the valueOf method of the String class**

**System.out.println("\nUsing valueOf method:");**

**System.out.println("Byte: " + String.valueOf(byteValue));**

**System.out.println("Short: " + String.valueOf(shortValue));**

**System.out.println("Integer: " + String.valueOf(intValue));**

**System.out.println("Long: " + String.valueOf(longValue));**

**System.out.println("Float: " + String.valueOf(floatValue));**

**System.out.println("Double: " + String.valueOf(doubleValue));**

**System.out.println("Boolean: " + String.valueOf(boolValue));**

**System.out.println("Character: " + String.valueOf(charValue));**

**}**

**}**

**Output:**

**Using toString method:**

**Byte: 10**

**Short: 20**

**Integer: 42**

**Long: 100000**

**Float: 3.14**

**Double: 3.14159**

**Boolean: true**

**Character: A**

**Using valueOf method:**

**Byte: 10**

**Short: 20**

**Integer: 42**

**Long: 100000**

**Float: 3.14**

**Double: 3.14159**

**Boolean: true**

**Character: A**

# 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).

**public class Q9 {**

**// Instance variables**

**byte instanceByte;**

**short instanceShort;**

**int instanceInt;**

**long instanceLong;**

**float instanceFloat;**

**double instanceDouble;**

**boolean instanceBoolean;**

**char instanceChar;**

**// Static variables**

**static byte staticByte;**

**static short staticShort;**

**static int staticInt;**

**static long staticLong;**

**static float staticFloat;**

**static double staticDouble;**

**static boolean staticBoolean;**

**static char staticChar;**

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

**// Create an object to access instance variables**

**Q9 obj = new Q9();**

**// Print default values of instance variables**

**System.out.println("Instance Variable Default Values:");**

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

**System.out.println("Short: " + obj.instanceShort);**

**System.out.println("Int: " + obj.instanceInt);**

**System.out.println("Long: " + obj.instanceLong);**

**System.out.println("Float: " + obj.instanceFloat);**

**System.out.println("Double: " + obj.instanceDouble);**

**System.out.println("Boolean: " + obj.instanceBoolean);**

**System.out.println("Char: [" + obj.instanceChar + "]");**

**// Print default values of static variables**

**System.out.println("\nStatic Variable Default Values:");**

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

**System.out.println("Short: " + staticShort);**

**System.out.println("Int: " + staticInt);**

**System.out.println("Long: " + staticLong);**

**System.out.println("Float: " + staticFloat);**

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

**System.out.println("Boolean: " + staticBoolean);**

**System.out.println("Char: [" + staticChar + "]");**

**}**

**}**

**Output:**

**Instance Variable Default Values:**

**Byte: 0**

**Short: 0**

**Int: 0**

**Long: 0**

**Float: 0.0**

**Double: 0.0**

**Boolean: false**

**Char: []**

**Static Variable Default Values:**

**Byte: 0**

**Short: 0**

**Int: 0**

**Long: 0**

**Float: 0.0**

**Double: 0.0**

**Boolean: false**

**Char: []**

# **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 Q10 {**

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

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

**char operator = args[1].charAt(0);**

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

**switch (operator) {**

**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("Enter valid operator");**

**break;**

**}**

**}**

**}**

**Output:**

**PS D:\Cdac\OOP Java\Assignments\Sandeep sir's Assignments\Assignment 2 codes> 11 - 10**

**1**

**PS D:\Cdac\OOP Java\Assignments\Sandeep sir's Assignments\Assignment 2 codes> 11 \* 1**

**11**

**PS D:\Cdac\OOP Java\Assignments\Sandeep sir's Assignments\Assignment 2 codes> 11 \* 2**

**22**

**PS D:\Cdac\OOP Java\Assignments\Sandeep sir's Assignments\Assignment 2 codes>**