#### ****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) ).

**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)).

**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").

**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)).

**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)).

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

**Program :-**

**package Day3;**

**public class Que1 {**

**public static void m1() {**

**boolean s = true;**

**String s1 = Boolean.*toString*(s);**

**System.*out*.println(s1);**

**}**

**public static void m2() {**

**String s="true";**

**boolean s1= Boolean.*parseBoolean*(s);**

**System.*out*.println(s1);**

**}**

**public static void m3() {**

**String s = "1";**

**boolean s1 = "1".equals(s);**

**System.*out*.println(s1);**

**}**

**public static void m4()**

**{**

**boolean s= true;**

**Boolean ws= Boolean.*valueOf*(s); // Wrapper class boolean to Boolean**

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

**}**

**public static void m5() {**

**String s= "true";**

**Boolean ws= Boolean.*valueOf*(s); // W class string to Boolean**

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

**}**

**public static void m6() {**

**boolean s= true;**

**int n= s ? 1 : 0 ;**

**System.*out*.println("boolean to int "+ n);**

**}**

**public static void m7() {**

**boolean b = false;**

**double d = b ? 1.0 : 0.0;**

**System.*out*.println("boolean to double "+ d);**

**}**

**public static void m8() {**

**boolean b = false;**

**long l = b ? 1L : 0L;**

**System.*out*.println("boolean to long "+ l);**

**}**

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

***m1*();**

***m2*();**

***m3*();**

***m4*();**

***m5*();**

***m6*();**

***m7*();**

***m8*();**

**}**

**}**

**Output**:-

#### true

#### true

#### true

#### true

#### true

#### boolean to int 1

#### boolean to double 0.0

#### boolean to long 0

#### ****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).

**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).

**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)).

**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)).

**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).

**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)).

**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)).

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

**Answer:-**

**package Day3;**

**public class Que2 {**

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

**int b= Byte.*BYTES*;**

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

**byte b1= Byte.*MIN\_VALUE*;**

**byte b2= Byte.*MAX\_VALUE*;**

**System.*out*.println("Min value of b1:" + b1 +"\nMax Value of b2: "+ b2);**

**byte n= 100;**

**String s = Byte.*toString*(n);**

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

**String s1 = "50";**

**byte b3= Byte.*parseByte*(s1);**

**System.*out*.println(b3);**

**// public static void sushil5() {**

**// String s = "Sushil7787";**

**// byte b = Byte.parseByte(s);**

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

**//**

**byte c = 20;**

**Byte b4=Byte.*valueOf*(c);**

**System.*out*.println(b4);**

**String c1= "10";**

**Byte b5= Byte.*valueOf*(c1);**

**System.*out*.println(b5);**

**}**

**}**

**// public static void main(String [] sushil) {**

**//**

**// sushil1();**

**//**

**// sushil2();**

**//**

**// sushil3();**

**//**

**// sushil4();**

**//**

**// // sushil5();**

**//**

**// sushil6();**

**//**

**// sushil7();**

**//**

**// }**

**//}**

**Output:-**

**1**

**Min value of b1:-128**

**Max Value of b2: 127**

**100**

**50**

**20**

**10**

**Q.7)**

**package Day3;**

**public class Que5 {**

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

**System.*out*.println("This are implicity Casting");**

**byte b = 100;**

**int a= b;**

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

**byte b1 = 50;**

**double d= b1;**

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

**byte b2= 80;**

**float f = b2;**

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

**byte b3= 30;**

**long l = b3;**

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

**System.*out*.println("\nThis are Explicity Casting");**

**int i = 20;**

**byte a1 =(byte)i;**

**System.*out*.println(a1);**

**double d1 = 20.30;**

**byte a2= (byte)d1;**

**System.*out*.println(a2);**

**float f1 = 30.3f;**

**byte a3= (byte) f1;**

**System.*out*.println(a3);**

**long l1 = 86;**

**byte a4= (byte)l1;**

**System.*out*.println(a4);**

**}**

**}**

**Output :-**

#### This are implicity Casting

#### 100

#### 50.0

#### 80.0

#### 30

#### This are Explicity Casting

#### 20

#### 20

#### 30

#### 86

#### ****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.

**Answer:-**

**package Day3;**

**public class Que4 {**

**public static void sushil1(){**

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

**}**

**public static void sushil2(){**

**System.*out*.println(Short.*MIN\_VALUE* );**

**System.*out*.println(Short.*MAX\_VALUE* );**

**}**

**public static void sushil3(){**

**Short s = 7787;**

**String s1= Short.*toString*(s);**

**System.*out*.println(s1);**

**}**

**public static void sushil4(){**

**String a= "8657";**

**Short a1= Short.*parseShort*(a);**

**System.*out*.println(a1);**

**}**

**// public static void sushil5(){**

**// String s="Sushil123";**

**// Short s1= Short.parseShort(s);**

**// System.out.println(s1);**

**// }**

**public static void sushil6(){**

**Short b = 1234;**

**Short b1= Short.*valueOf*(b);**

**System.*out*.println(b1);**

**}**

**public static void sushil7(){**

**String c = "9640";**

**Short c1 = Short.*valueOf*(c);**

**System.*out*.println(c1);**

**}**

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

***sushil1*();**

***sushil2*();**

***sushil3*();**

***sushil4*();**

**// sushil5();**

***sushil6*();**

***sushil7*():**

**}**

**}**

#### ****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).

**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).

**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)).

**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)).

**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).

**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)).

**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)).

**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)).

**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)).

**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)).

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

**Program :-**

**package examples;**

**public class Que1 {**

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

**System.*out*.println("interger byte: " + Integer.*BYTES*);**

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

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

**int n = 123456;**

**String s = Integer.*toString*(n);**

**System.*out*.println("Int to String: " + s);**

**String s1 = "123456";**

**int n1 = Integer.*parseInt*(s1);**

**System.*out*.println("String to int: " + n1);**

**//**

**// String s2 = "Ab12Cd3";**

**// int n2= Integer.parseInt(s2);**

**// System.out.println("String to int: " + n2);**

**int n3 = 123456;**

**Integer i = Integer.*valueOf*(n3);**

**System.*out*.println("Int to Integer wrapper: " + i);**

**String s4 = "123456";**

**Integer i1 = Integer.*valueOf*(s4);**

**System.*out*.println("String to Integer wrapper: " + i1);**

**int a = 10;**

**int b = 20;**

**int sum = Integer.*sum*(a, b);**

**System.*out*.println("Sum of 10 and 20: " + sum);**

**int a1 = 10;**

**int b1 = 20;**

**int min = Integer.*min*(a1, b1);**

**int max = Integer.*max*(a1, b1);**

**System.*out*.println("Min of 10 and 20: " + min);**

**System.*out*.println("Max of 10 and 20: " + max);**

**int nu = 7;**

**String bi = Integer.*toBinaryString*(nu);**

**String oc= Integer.*toOctalString*(nu);**

**String he = Integer.*toHexString*(nu);**

**System.*out*.println("Bi 7: " + bi);**

**System.*out*.println("Oc 7: " + oc);**

**System.*out*.println("He 7: " + he);**

**}**

**}**

**Output;-**

**interger byte: 4**

**Min value: -2147483648**

**Max value: 2147483647**

**Int to String: 123456**

**String to int: 123456**

**Int to Integer wrapper: 123456**

**String to Integer wrapper: 123456**

**Sum of 10 and 20: 30**

**Min of 10 and 20: 10**

**Max of 10 and 20: 20**

**Bi 7: 111**

**Oc 7: 7**

**He 7: 7**

#### ****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.

**Program:**-

**package examples;**

**public class Que2 {**

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

**System.*out*.println("Number of bytes " + Long.*BYTES*);**

**System.*out*.println("Min: " + Long.*MIN\_VALUE*);**

**System.*out*.println("Max: " + Long.*MAX\_VALUE*);**

**long n= 123456789L;**

**String s = Long.*toString*(n);**

**System.*out*.println("Long to String: " + s);**

**String s1= "123456789";**

**long n1 = Long.*parseLong*(s1);**

**System.*out*.println("String to long: " + n1);**

**// String s2 = "Ab12Cd3";**

**// long n2 = Long.parseLong(s2);**

**// System.out.println("String to long: " + n2);**

**//**

**long n3 = 123456789L;**

**Long l = Long.*valueOf*(n3);**

**System.*out*.println("Long to Long wrapper: " + l);**

**String s3 = "123456789";**

**Long l2 = Long.*valueOf*(s3);**

**System.*out*.println("String to Long wrapper: " + l2);**

**}**

**}**

#### ****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.

**Program:-**

**package examples;**

**public class Que3 {**

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

**System.*out*.println("Number of bytes: " + Float.*BYTES*);**

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

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

**float n = 123.456f;**

**String s = Float.*toString*(n);**

**System.*out*.println("Float to String: " + s);**

**String sr = "123.456";**

**float s1 = Float.*parseFloat*(sr);**

**System.*out*.println("String to float: " + s1);**

**String s2 = "Ab12Cd3";**

**float s3 = Float.*parseFloat*(s2);**

**System.*out*.println("String to float: " + s3);**

**Float n1 = Float.*valueOf*(n);**

**System.*out*.println("Float to Float wrapper: " + n1);**

**// Float fc = Float.valueOf(f);**

**// System.out.println("String to Float wrapper: " + fc);**

**float a = 112.3f;**

**float b = 984.5f;**

**float sum = Float.*sum*(a, b);**

**System.*out*.println("Sum of 112.3 and 984.5: " + sum);**

**float c = 112.2f;**

**float d = 556.6f;**

**float min = Float.*min*(c, d);**

**float max = Float.*max*(c, d);**

**System.*out*.println("Minimum of 112.2 and 556.6: " + min);**

**System.*out*.println("Maximum of 112.2 and 556.6: " + max);**

**float negativeNumber = -25.0f;**

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

**System.*out*.println("Square root of -25.0f: " + sqrt);**

**float zero1 = 0.0f;**

**float zero2 = 0.0f;**

**float dr = zero1 / zero2;**

**System.*out*.println("0.0f divided by 0.0f: " + dr);**

**float f2 = 130.7f;**

**byte bv = (byte) f2;**

**System.*out*.println("Float to byte: " + bv);**

**float fs= 32768.7f;**

**short shortValue = (short) fs;**

**System.*out*.println("Float to short: " + shortValue);**

**float f3 = 2147483648.7f;**

**int iv = (int) f3;**

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

**float f4 = 9223372036854775807f;**

**long lv = (long) f4;**

**System.*out*.println("Float to long: " + lv);**

**float f5 = 123456.789f;**

**double dv = f5;**

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

**byte b1 = 100;**

**float fb = b1;**

**System.*out*.println("Byte to float: " + fb);**

**short s4 = 1000;**

**float fd = s4;**

**System.*out*.println("Short to float: " + fd);**

**}**

**}**

**Output:-**

**Number of bytes 8**

**Min: -9223372036854775808**

**Max: 9223372036854775807**

**Long to String: 123456789**

**String to long: 123456789**

**Long to Long wrapper: 123456789**

**String to Long wrapper: 123456789**

**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.**

**Program:-**

**package examples;**

**public class Que4 {**

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

**System.*out*.println("Number of bytes : " + Double.*BYTES*);**

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

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

**double number = 123.456;**

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

**System.*out*.println("String representation of the double value: " + numberAsString);**

**String strNumber = "123.456";**

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

**System.*out*.println("Double value from string: " + parsedNumber);**

**String invalidStrNumber = "Ab12Cd3";**

**try {**

**double invalidNumber = Double.*parseDouble*(invalidStrNumber);**

**System.*out*.println("Double value from string: " + invalidNumber);**

**} catch (NumberFormatException e) {**

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

**}**

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

**System.*out*.println("Double wrapper class from double value: " + wrapperFromDouble);**

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

**System.*out*.println("Double wrapper class from string: " + wrapperFromString);**

**double num1 = 112.3;**

**double num2 = 984.5;**

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

**System.*out*.println("Sum of the two double values: " + sum);**

**double num3 = 112.2;**

**double num4 = 556.6;**

**double min = Double.*min*(num3, num4);**

**double max = Double.*max*(num3, num4);**

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

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

**double negativeNumber = -25.0;**

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

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

**double zero1 = 0.0;**

**double zero2 = 0.0;**

**double divisionResult = zero1 / zero2;**

**System.*out*.println("Result of dividing 0.0 by 0.0: " + divisionResult);**

**}**

**}**

**Output:-**

**Number of bytes : 8**

**Minimum value : 4.9E-324**

**Maximum value : 1.7976931348623157E308**

**String representation of the double value: 123.456**

**Double value from string: 123.456**

**Error: For input string: "Ab12Cd3"**

**Double wrapper class from double value: 123.456**

**Double wrapper class from string: 123.456**

**Sum of the two double values: 1096.8**

**Minimum value: 112.2**

**Maximum value: 556.6**

**Square root of -25.0 is: NaN**

**Result of dividing 0.0 by 0.0: NaN**

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()).

**Program:-**

**package examples;**

**public class Que5 {**

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

**byte byteValue = 10;**

**String byteString = Byte.*toString*(byteValue);**

**System.*out*.println("Byte value as String: " + byteString);**

**short shortValue = 100;**

**String shortString = Short.*toString*(shortValue);**

**System.*out*.println("Short value as String: " + shortString);**

**int intValue = 1000;**

**String intString = Integer.*toString*(intValue);**

**System.*out*.println("Int value as String: " + intString);**

**long longValue = 10000L;**

**String longString = Long.*toString*(longValue);**

**System.*out*.println("Long value as String: " + longString);**

**float floatValue = 10.5f;**

**String floatString = Float.*toString*(floatValue);**

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

**double doubleValue = 100.99;**

**String doubleString = Double.*toString*(doubleValue);**

**System.*out*.println("Double value as String: " + doubleString);**

**char charValue = 'A';**

**String charString = Character.*toString*(charValue);**

**System.*out*.println("Char value as String: " + charString);**

**boolean booleanValue = true;**

**String booleanString = Boolean.*toString*(booleanValue);**

**System.*out*.println("Boolean value as String: " + booleanString);**

**}**

**}**

**Output:-**

**Number of bytes : 8**

**Minimum value : 4.9E-324**

**Maximum value : 1.7976931348623157E308**

**String representation of the double value: 123.456**

**Double value from string: 123.456**

**Error: For input string: "Ab12Cd3"**

**Double wrapper class from double value: 123.456**

**Double wrapper class from string: 123.456**

**Sum of the two double values: 1096.8**

**Minimum value: 112.2**

**Maximum value: 556.6**

**Square root of -25.0 is: NaN**

**Result of dividing 0.0 by 0.0: NaN**

**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).**

**Program:-**

**package** examples;

**public** **class** Que6 {

**byte** ib;

**short** is;

**int** it;

**long** il;

**float** iff;

**double** id;

**char** ic;

**boolean** ibb;

**static** **byte** *sb*;

**static** **short** *ss*;

**static** **int** *si*;

**static** **long** *sl*;

**static** **float** *sf*;

**static** **double** *sd*;

**static** **char** *sc*;

**static** **boolean** *sbb*;

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

Que6 example = **new** Que6();

System.***out***.println("Default values of instance variables:");

System.***out***.println("byte: " + example.ib);

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

System.***out***.println("int: " + example.it);

System.***out***.println("long: " + example.il);

System.***out***.println("float: " + example.iff);

System.***out***.println("double: " + example.id);

System.***out***.println("char: " + example.ic);

System.***out***.println("boolean: " + example.ibb);

System.***out***.println("\nDefault values of static variables:");

System.***out***.println("byte: " + Que6.*sb*);

System.***out***.println("short: " + Que6.*ss*);

System.***out***.println("int: " + Que6.*si*);

System.***out***.println("long: " + Que6.*sc*);

System.***out***.println("float: " + Que6.*sf*);

System.***out***.println("double: " + Que6.*sd*);

System.***out***.println("char: " + Que6.*sc*);

System.***out***.println("boolean: " + Que6.*sbb*);

} }

Output:-

Number of bytes : 8

Minimum value : 4.9E-324

Maximum value : 1.7976931348623157E308

String representation of the double value: 123.456

Double value from string: 123.456

Error: For input string: "Ab12Cd3"

Double wrapper class from double value: 123.456

Double wrapper class from string: 123.456

Sum of the two double values: 1096.8

Minimum value: 112.2

Maximum value: 556.6

Square root of -25.0 is: NaN

Result of dividing 0.0 by 0.0: NaN

**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).

Prpgram:-

**package** examples;

**public** **class** Que7 {

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

// Ensure there are exactly three arguments

**if** (args.length != 3) {

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

**return**;}

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

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

String operator = args[1];

**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((**double**) num1 / num2); // Use double for division to handle non-integer results

} **else** {

System.***out***.println("Error: Division by zero");}

**break**;

**default**:

System.***out***.println("Error: Invalid operator. Use +, -, \*, or /.");

**break**;

} } }