**Note**: Consider the following before starting the assignment:

* A **static field** declared inside a class is called a **class-level variable**. To access this variable, use the class name and the dot operator (e.g., Integer.MAX\_VALUE).
* A **static method** defined inside a class is called a **class-level method**. To access this method, use the class name and the dot operator (e.g., Integer.parseInt()).
* When accessing static members within the same class, you do not need to use the class name.

#### ****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.

=> public final class **Boolean**

extends [Object](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)

implements [Serializable](https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html), [Comparable](https://docs.oracle.com/javase/8/docs/api/java/lang/Comparable.html)<[Boolean](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html)>

The Boolean class wraps a value of the primitive type boolean in an object. An object of type Boolean contains a single field whose type is boolean.

In addition, this class provides many methods for converting a boolean to a String and a String to a boolean, as well as other constants and methods useful when dealing with a boolean.

|  |  |
| --- | --- |
| **Fields** | |
| **Modifier and Type** | **Field and Description** |
| static [**Boolean**](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html) | [**FALSE**](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html#FALSE)  The Boolean object corresponding to the primitive value false. |
| static [**Boolean**](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html) | [**TRUE**](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html#TRUE)  The Boolean object corresponding to the primitive value true. |
| static [**Class**](https://docs.oracle.com/javase/8/docs/api/java/lang/Class.html)<[**Boolean**](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html)> | [**TYPE**](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html#TYPE)  The Class object representing the primitive type boolean. |

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

**public class demoboolean{**

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

**{**

**boolean status=true;**

**String strbool = Boolean.toString( status);**

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

**}**

**}**

**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 pboolean{**

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

**{**

**String strStatus =new String("true");**

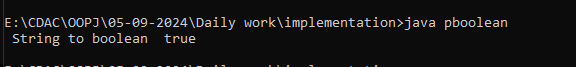
**boolean boolstatus=Boolean.parseBoolean(strStatus);**

**System.out.println(" String to boolean "+boolstatus);**

**}**

**}**

**Output:**



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

**public class Qd{**

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

**{**

**String strStatus=new String("1");**

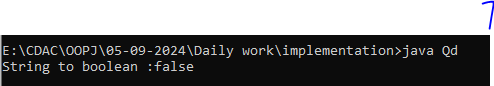
**Boolean bool=Boolean.parseBoolean(strStatus);**

**System.out.println("String to boolean :"+bool);**

**}**

**}**

**Output:**



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

public class Qe{

public static void main(String args[])

{

Boolean status = true;

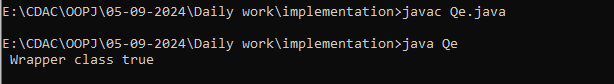
Boolean statusWrapper = Boolean.valueOf(status);

System.out.println(" Wrapper class "+statusWrapper);

}

}

Output:



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

**public class Qf{**

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

**{**

**String status = "true";**

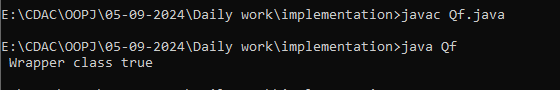
**Boolean statusWrapper = Boolean.valueOf(status);**

**System.out.println(" Wrapper class "+statusWrapper);**

**}**

**}**

**Output:**



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

public class Qg{

public static void main(String args [])

{

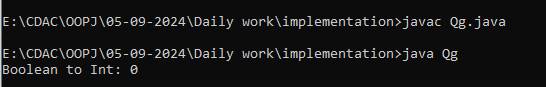
boolean status=false;

int strbool = (status) ? 1 :0;

System.out.println("Boolean to Int: "+strbool);

}

}



#### ****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 bytess{**

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

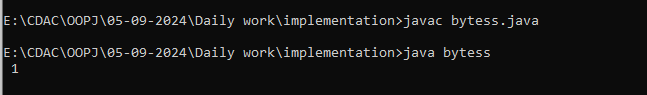
**{**

**System.out.println(" "+Byte.BYTES);**

**}**

**}**

**Output:**



**c.** Write a program to find the minimum and maximum values of byte using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Byte.MIN\_VALUE and Byte.MAX\_VALUE).

public class Q2c{

public static void main(String args[])

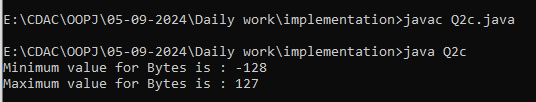
{

System.out.println("Minimum value for Bytes is : "+Byte.MIN\_VALUE );

System.out.println("Maximum value for Bytes is : "+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)).

**public class Q2d{**

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

**{**

**byte number= 1;**

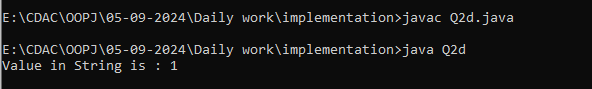
**String str=Byte.toString(number);**

**System.out.println("Value in String is : "+str );**

**}**

**}**

**Output:**



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

public class Q2e{

public static void main(String args[])

{

String strNumber=new String("42");

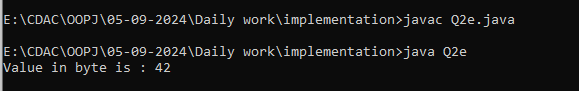
byte bytenum=Byte.parseByte(strNumber);

System.out.println("Value in byte is : "+bytenum );

}

}

Output:



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

public class Q2f{

public static void main(String args[])

{

String strNumber=new String("Ab12Cd3");

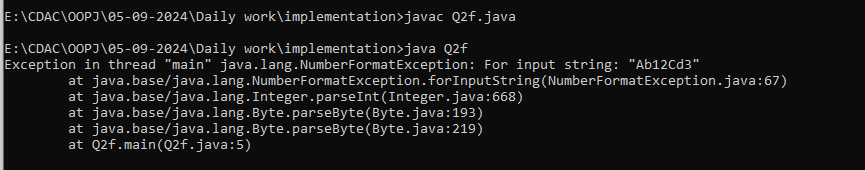
byte bytenum=Byte.parseByte(strNumber);

System.out.println("Value in byte is : "+bytenum );

}

}

Output:



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

**public class Q2g{**

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

**{**

**byte number=2;**

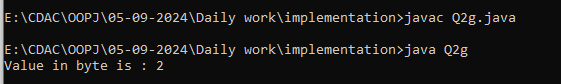
**byte bytenum=Byte.valueOf(number);**

**System.out.println("Value in byte is : "+number );**

**}**

**}**

**Output:**



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

public class Q2h{

public static void main(String args[])

{

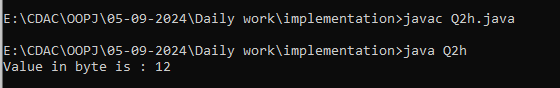
String strnumber=new String("12");

byte bytenum=Byte.valueOf(strnumber);

System.out.println("Value in byte is : "+strnumber );

}

} .



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

public class Q2i{

public static void main(String args[])

{

char dnumber=new char("a");

byte bytenum=Byte.valueOf(dnumber);

System.out.println("Value in byte is : "+dnumber );

}

}

#### ****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).

public class Q3b{

public static void main(String args[])

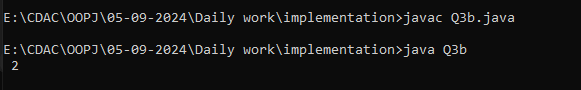
{

System.out.println(" "+Short.BYTES);

}

}

Output:



**c.** Write a program to find the minimum and maximum values of short using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Short.MIN\_VALUE and Short.MAX\_VALUE).

public class Q3c{

public static void main(String args[])

{

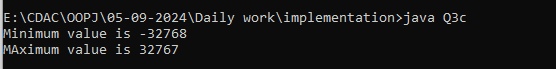
System.out.println("Minimum value is "+Short.MIN\_VALUE);

System.out.println("MAximum value is "+Short.MAX\_VALUE);

}

}

Output:



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

public class Q3d{

public static void main(String args[])

{

short number= 1;

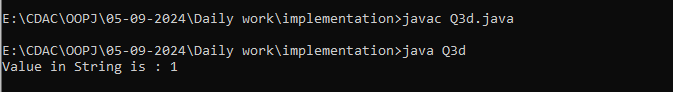
String str=Short.toString(number);

System.out.println("Value in String is : "+str );

}

}

Output:



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

public class Q3e{

public static void main(String[] args)

{

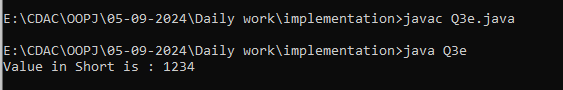
String strNumber = "1234";

short number = Short.parseShort(strNumber);

System.out.println("Value in short is : "+number );

}

}



**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 Q3f{

public static void main(String args[])

{

String strNumber = "Ab12Cd3";

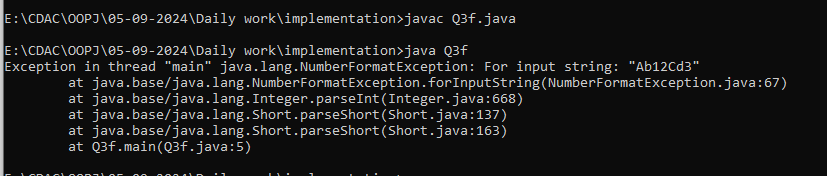
short number = Short.parseShort(strNumber);

System.out.println("Value in short is : "+number );

}

}

Output:



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

public class Q3g{

public static void main(String args[])

{

short number=2;

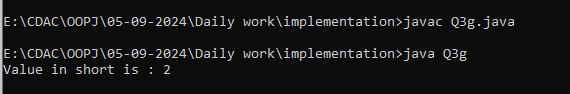
short num=Short.valueOf(number);

System.out.println("Value in short is : "+num );

}

}

Output:



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

**public class Q3h{**

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

**{**

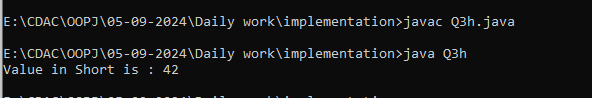
**String strnumber="42";**

**short shortnum=Short.valueOf(strnumber);**

**System.out.println("Value in Short is : "+strnumber );**

**}**

**}**



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

public class Q3i{

public static void main(String args[])

{

short strnumber=42;

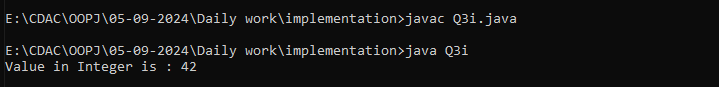
int intnum=Integer.valueOf(strnumber);

System.out.println("Value in Integer is : "+strnumber );

}

}

Output:

1. 

#### ****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 Q4b{**

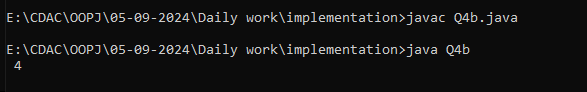
**public static void main(String args[])**

**{**

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

public class Q4c{

public static void main(String args[])

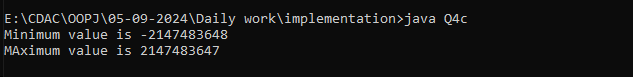
{

System.out.println("Minimum value is "+Integer.MIN\_VALUE);

System.out.println("MAximum value is "+Integer.MAX\_VALUE);

}

}



**d.** Declare a method-local variable number of type int with some value and convert it to a String using the to public class Q4d{

public static void main(String args[])

{

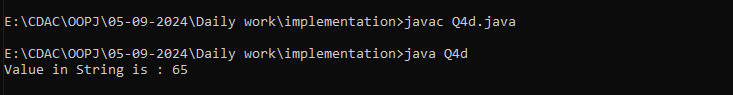
int number=65;

String str=Integer.toString(number);

System.out.println("Value in String is : "+str );

}

} String 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)).

**public class Q4e{**

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

**{**

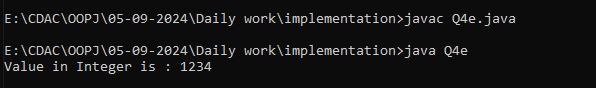
**String strNumber = "1234";**

**Integer number = Integer.parseInt(strNumber);**

**System.out.println("Value in Integer is : "+number );**

**}**

**}**



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

public class Q4f{

public static void main(String args[])

{

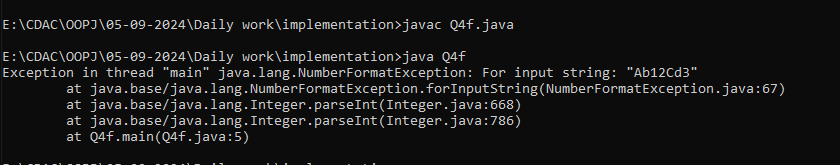
String strNumber = "Ab12Cd3";

int number = Integer.parseInt (strNumber);

System.out.println("Value in Short is : "+number );

}

}



**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 Q4g{

public static void main(String args[])

{

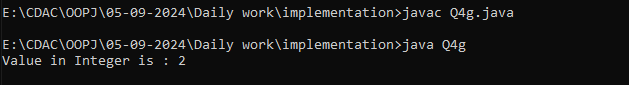
int number=2;

int num=Integer.valueOf(number);

System.out.println("Value in Integer is : "+num );

}

}



**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 Q4h{

public static void main(String args[])

{

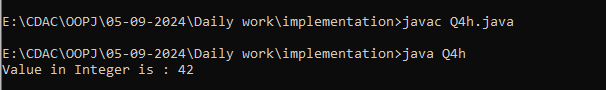
String strnumber="42";

int shortnum=Integer.valueOf(strnumber);

System.out.println("Value in Integer is : "+strnumber );

}

}



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

**public class Q4i{**

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

**{**

**int a = 10;**

**int b = 20;**

**System.out.println("Sum is : " );**

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

**}**

**}**

**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 Q4i{

public static void main(String args[])

{

int a = 10;

int b = 20;

int c= Integer.min(a, b);

int d= Integer.max(a, b);

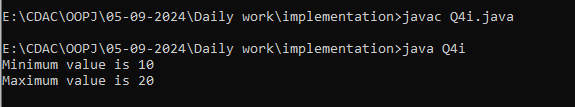
System.out.println("Minimum value is "+c );

System.out.println("Maximum value is "+d );

}

}

Output:



**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.

#### ****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).

**public class Q5b{**

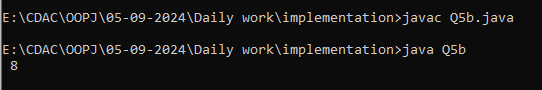
**public static void main(String args[])**

**{**

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

**}**

**}**



**c.** Write a program to find the minimum and maximum values of long using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Long.MIN\_VALUE and Long.MAX\_VALUE).

public class Q5c{

public static void main(String args[])

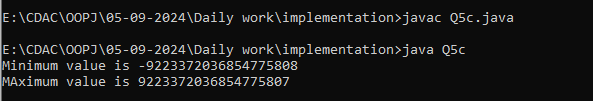
{

System.out.println("Minimum value is "+Long.MIN\_VALUE);

System.out.println("MAximum value is "+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)).

**public class Q5d{**

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

**{**

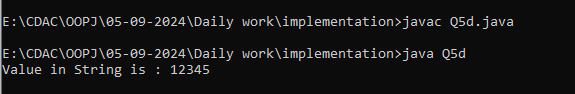
**Long number=65l;**

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

**System.out.println("Value in String is : "+str );**

**}**

**}**



**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 Q5e{

public static void main(String[] args)

{

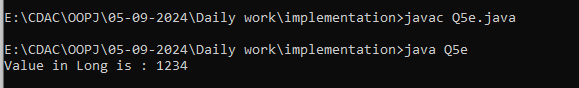
String strNumber = "1234";

long number = Long.parseLong(strNumber);

System.out.println("Value in Long is : "+number );

}

}

.

**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 Q5f{**

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

**{**

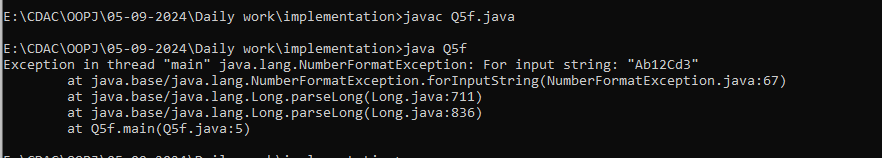
**String strNumber = "Ab12Cd3";**

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

**System.out.println("Value in Long is : "+number );**

**}**

**}**



**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 Q5g{**

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

**{**

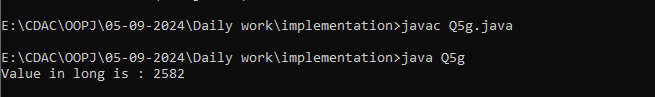
**long number=2582l;**

**long num=Long.valueOf(number);**

**System.out.println("Value in long is : "+num );**

**}**

**}**

**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 Q5h{**

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

**{**

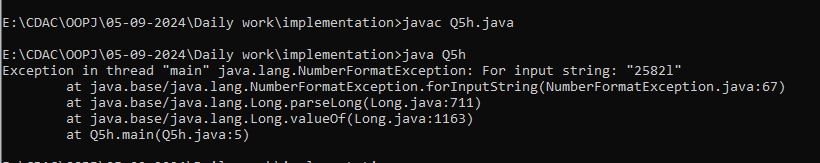
**String strnumber="2582l";**

**long num=Long.valueOf(strnumber);**

**System.out.println("Value in long is : "+num );**

**}**

**}**



**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)).

**class Q5i{**

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

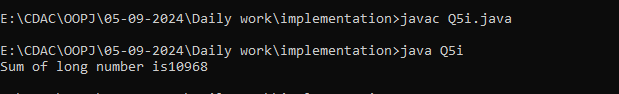
**long num1=1123l;**

**long num2=9845l;**

**System.out.println("Sum of long number is"+Long.sum(num1,num2));**

**}**

**}**



**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)).

**class Q5j{**

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

**long num1=1122l;**

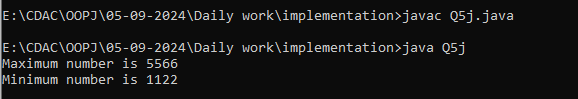
**long num2=5566l;**

**System.out.println("Maximum number is "+Long.max(num1,num2));**

**System.out.println("Minimum number is "+Long.min(num1,num2));**

**}**

**}**



**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.

#### ****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).

public class Q6b{

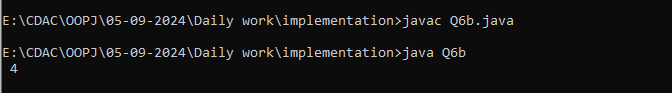
public static void main(String args[])

{

System.out.println(" "+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).

public class Q6c{

public static void main(String args[])

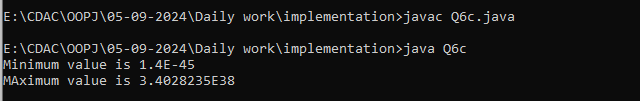
{

System.out.println("Minimum value is "+Float.MIN\_VALUE);

System.out.println("MAximum value is "+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)).

public class Q6d{

public static void main(String args[])

{

float number=12345f;

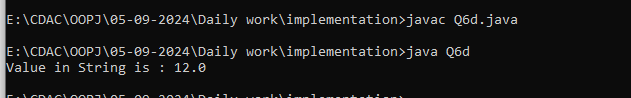
String str=Float.toString(number);

System.out.println("Value in String is : "+str );

}

}

Output:



**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 Q6e{

public static void main(String[] args)

{

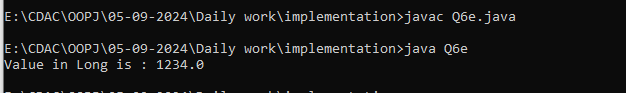
String strNumber = "1234";

float number = Float.parseFloat(strNumber);

System.out.println("Value in Long is : "+number );

}

}



**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 Q6f{**

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

**{**

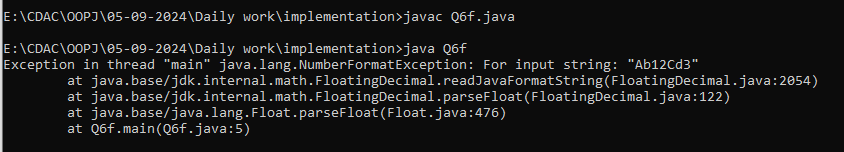
**String strNumber = "Ab12Cd3";**

**float number = Float.parseFloat (strNumber);**

**System.out.println("Value in Float is : "+number );**

**}**

**}**



**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 Q6g{**

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

**{**

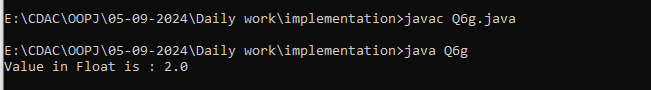
**float number=2f;**

**float num=Float.valueOf(number);**

**System.out.println("Value in Float is : "+num );**

**}**

**}**



**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 Q6h{

public static void main(String args[])

{

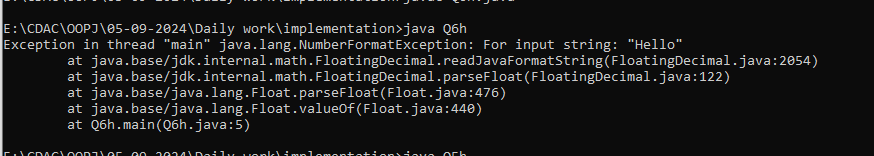
String number="258745";

float num=Float.valueOf(number);

System.out.println("Value in Float is : "+num );

}

}



1. 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 Q6i{

public static void main(String args[])

{

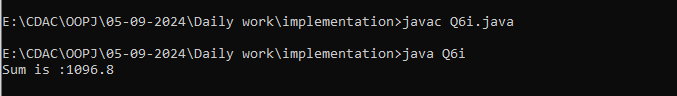
float num1=112.3f;

float num2=984.5f;

System.out.println("Sum is :"+Float.sum(num1,num2));

}

}



**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 Q6j{

public static void main(String args[])

{

float num1=112.2f;

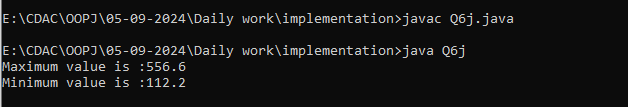
float num2=556.6f;

System.out.println("Maximum value is :"+Float.max(num1,num2));

System.out.println("Minimum value is :"+Float.min(num1,num2));

}

}



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

**public class Q6k{**

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

**{**

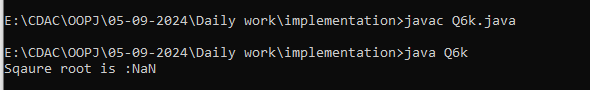
**float num1=-25.0f;**

**float result=(float)Math.sqrt(num1);**

**System.out.println("Sqaure root is :"+result);**

**}**

**}**



**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 Q6l{**

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

**{**

**float num1=0.0f;**

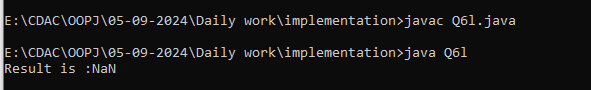
**float num2=0.0f;**

**float result=num1/num2;**

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

**}**

**}**



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

#### ****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).

public class Q7b{

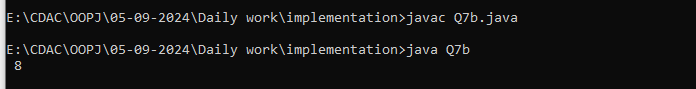
public static void main(String args[])

{

System.out.println(" "+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).

public class Q7c{

public static void main(String args[])

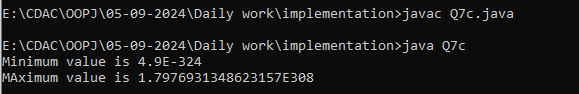
{

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

System.out.println("MAximum value is "+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)).

public class Q7d{

public static void main(String args[])

{

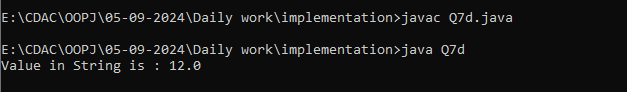
double number=12d;

String str=Double.toString(number);

System.out.println("Value in String is : "+str );

}

}



**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 Q7e{**

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

**{**

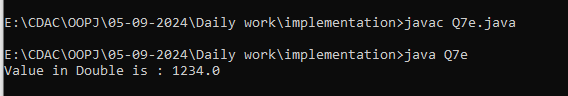
**String strNumber = "1234";**

**float number = Float.parseFloat(strNumber);**

**System.out.println("Value in Long is : "+number );**

**}**

**}**



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

public class Q7f{

public static void main(String args[])

{

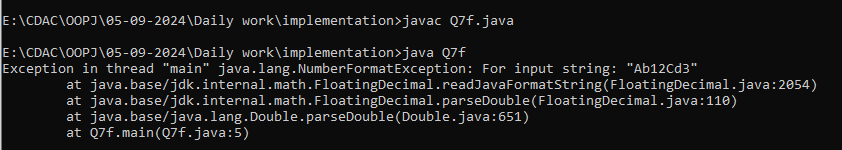
String strNumber = "Ab12Cd3";

double number = Double.parseDouble (strNumber);

System.out.println("Value in double is : "+number );

}

}



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

public class Q7g{

public static void main(String args[])

{

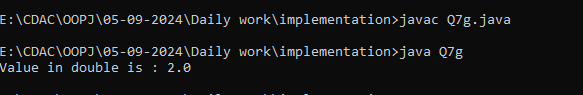
double number=2d;

double num=Double.valueOf(number);

System.out.println("Value in double is : "+num );

}

}



**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 Q7h{

public static void main(String args[])

{

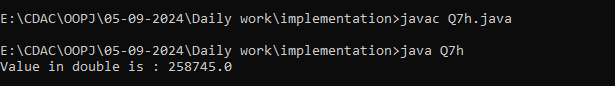
String number="258745";

double num=Double.valueOf(number);

System.out.println("Value in double is : "+num );

}

}



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

public class Q7i{

public static void main(String args[])

{

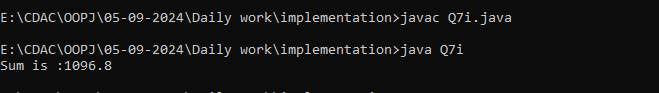
double num1=112.3d;

double num2=984.5d;

System.out.println("Sum is :"+Double.sum(num1,num2));

}

}



**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 Q7j{

public static void main(String args[])

{

double num1=112.2d;

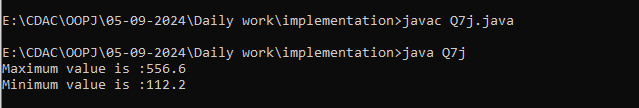
double num2=556.6d;

System.out.println("Maximum value is :"+Double.max(num1,num2));

System.out.println("Minimum value is :"+Double.min(num1,num2));

}

}



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

public class Q7k{

public static void main(String args[])

{

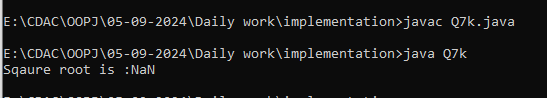
double num1=-25.0d;

double result=Math.sqrt(num1);

System.out.println("Sqaure root is :"+result);

}

}



**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 Q7l{

public static void main(String args[])

{

double num1=0.0d;

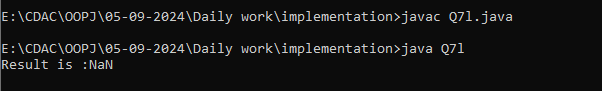
double num2=0.0d;

double result=num1/num2;

System.out.println("Result is :"+result);

}

}



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

public class Q7m{

public static void main(String args[])

{

double strnumber=42;

int intnum=Integer.valueOf(strnumber);

System.out.println("Value in Integer is : "+strnumber );

}

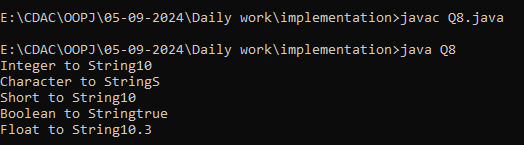
}

#### ****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()).

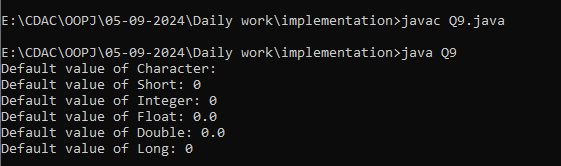
public class problem8  
{  
public static void main(String[]args)  
{  
int a=10;  
String ab=Integer.toString(a);  
String b=String.valueOf(ab);  
System.out.println("Integer to String"+b);  
  
char c='S';  
String cd=Character.toString(c);  
String d=String.valueOf(cd);  
System.out.println("Character to String"+d);  
  
short e=10;  
String ef=Short.toString(e);  
String f=String.valueOf(ef);  
System.out.println("Short to String"+f);  
  
boolean g=true;  
String gh=Boolean.toString(g);  
String h=String.valueOf(gh);  
System.out.println("Boolean to String"+h);  
  
float i=10.3f;  
String ij=Float.toString(i);  
String j=String.valueOf(ij);  
System.out.println("Float to String"+j);  
  
}  
  
  
}



#### ****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 problem9  
{  
static char a;  
static short b;  
static int c;  
static float d;  
static double e;  
static long f;  
  
public static void main(String[]args)  
{  
  
System.out.println("Default value of Character: "+a);  
System.out.println("Default value of Short: "+b);  
System.out.println("Default value of Integer: "+c);  
System.out.println("Default value of Float: "+d);  
System.out.println("Default value of Double: "+e);  
System.out.println("Default value of Long: "+f);  
  
}  
  
  
}



#### ****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 comline  
{  
public static void main(String args[])  
{  
  
int numa=Integer.parseInt(args[0]);  
int numb=Integer.parseInt(args[1]);  
//char c=Character.parseChar(args[2]);  
char ch=args[2].charAt(0);  
System.out.println("NUMBER 1 ="+numa);  
System.out.println("NUMBER 2 ="+numb);  
  
switch(ch)  
{  
case '+':  
int add=numa+numb;  
System.out.println("Addition of "+numa+"&"+numb+"is :"+add);  
break;  
case '-':  
int sub=numa-numb;  
System.out.println("Substraction of "+numa+"&"+numb+"is :"+sub);  
break;  
case '\*':  
int mul=numa\*numb;  
System.out.println("Multiplication of "+numa+"&"+numb+"is :"+mul);  
break;  
case '/':  
int div=numa/numb;  
System.out.println("Division of "+numa+"&"+numb+"is :"+div);  
break;  
default:  
System.out.println("Invalid input");  
}  
  
  
  
  
}  
}