#### ****CDAC MUMBAI****

#### ****MODULE 2 OOPJ****

#### ****ASSIGNMENT NO 2****

#### ****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.

**Answer:**

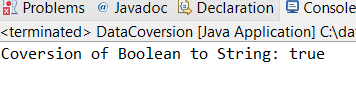
The java.lang.Boolean class wraps a value of the primitive type boolean in an object. An object of type Boolean contains a single field whose type is boolean.

* **Modifiers:** The Boolean class is final, meaning it cannot be subclassed. The fields TRUE and FALSE are public static final, representing the two possible values of Boolean.
* **Super types:** The Boolean class extends java.lang.Object.

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

**Program:**

**\*Using Text box for Code Insertion.**

Output:   


**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**boolean** type = **true**;

String typeStr = Boolean.*toString*(type);

System.***out***.println("Coversion of Boolean to String: " + typeStr);

}

}

**c.** Declare a method-local variable strStatus of type String with the value "true" and convert it to a boolean using the parseBoolean method. (Hint: Use Boolean.parseBoolean(String)).

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strStatus = "true";

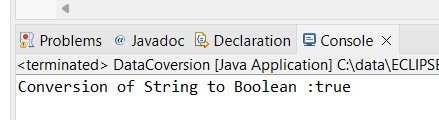
**boolean** type1 = Boolean.*parseBoolean*(strStatus);

System.***out***.println("Conversion of String to Boolean :" + type1);

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strStatus = "1";

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

System.***out***.println("Conversion of String 1 or 0 Boolean:" + type);

strStatus = "0";

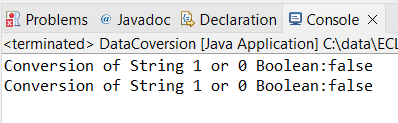
**boolean** type1 = Boolean.*parseBoolean*(strStatus);

System.***out***.println("Conversion of String 1 or 0 Boolean:" + type);

}

}

Output:



Observation : 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)).

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**boolean** status = **true**;

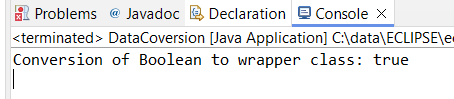
**boolean** type = Boolean.*valueOf*(status);

System.***out***.println("Conversion of Boolean to wrapper class: " + type);

}

}

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)).

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strStatus = "true";

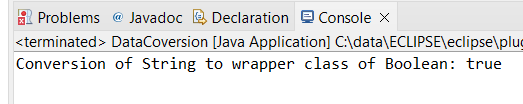
**boolean** type = Boolean.*valueOf*(strStatus);

System.***out***.println("Conversion of String to wrapper class of Boolean: " + type);

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strStatus = "false";

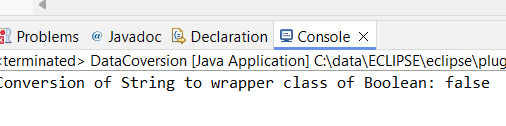
**boolean** type = Boolean.*valueOf*(strStatus);

System.***out***.println("Conversion of String to wrapper class of Boolean: " + type);

}

}

Output:



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

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

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

**program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

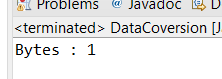
**byte** value = Byte.***BYTES***;

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

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**byte** value2 = Byte.***MAX\_VALUE***;

**byte** value3 = Byte.***MIN\_VALUE***;

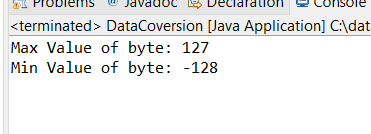
System.***out***.println("Max Value of byte: " + value2);

System.***out***.println("Min Value of byte: " + value3);

}

}

Output:



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

Program:

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**byte** type5 = 12;

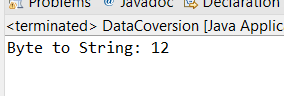
String str = Byte.*toString*(type5);

System.***out***.println("Byte to String: " + 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)).

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strNumber = "121";

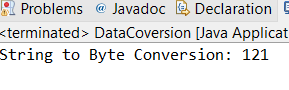
**byte** value4 = Byte.*parseByte*(strNumber);

System.***out***.println("String to Byte Conversion: " + value4);

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strNumber2="Ab12Cd3";

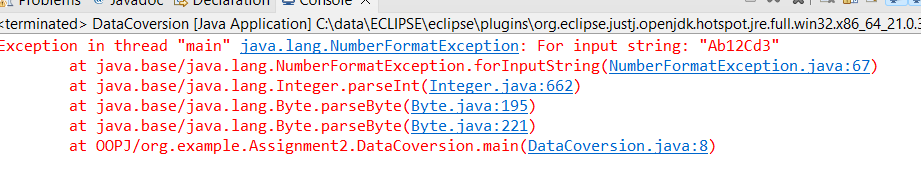
**byte** value5=Byte.*parseByte*(strNumber2);

System.***out***.println("String to Byte : "+value5);

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**byte** number = 12;

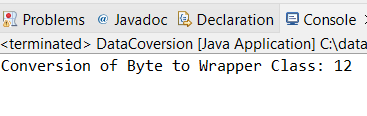
**byte** value6 = Byte.*valueOf*(number);

System.***out***.println("Conversion of Byte to Wrapper Class: " + value6);

}

}

Output:



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

Program:

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strNumber3 = "23";

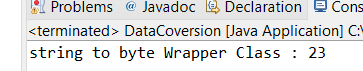
**byte** value7 = Byte.*valueOf*(strNumber3);

System.***out***.println("string to byte Wrapper Class : " + value7);

}

}

Output:



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

Program:

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**int** num = 12;

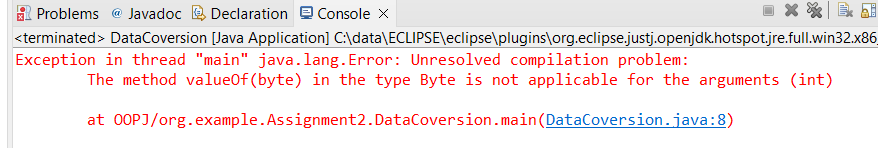
**byte** value8 = Byte.*valueOf*(num);

System.***out***.println("Intger to Byte: " + value8);

}

}

Output:



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

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

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

Program:

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

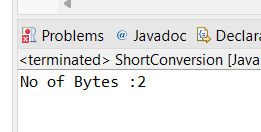
**short** value = Short.***BYTES***;

System.***out***.println("No of Bytes :" + value);

}

}

Output:



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

Program:

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

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

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

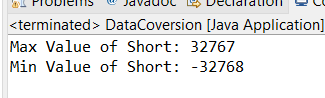
System.***out***.println("Max Value of Short: " + max);

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

}

}

Output:



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

Program:

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**short** number = 2345;

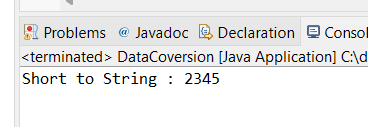
String str = Short.*toString*(number);

System.***out***.println("Short to String : " + 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)).

Program:

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strNumber2 = "12345";

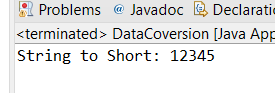
**short** value4 = Short.*parseShort*(strNumber2);

System.***out***.println("String to Short: " + value4);

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strNumber3 = "Ab12Cd3";

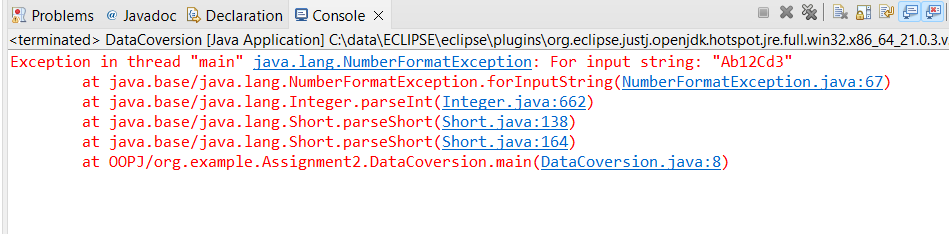
**short** value5 = Short.*parseShort*(strNumber3);

System.***out***.println("Character String into Short : " + value5);

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**short** num2 = 1234;

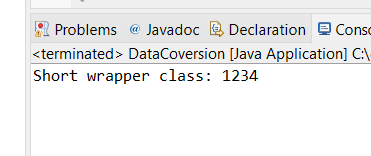
**short** value6 = Short.*valueOf*(num2);

System.***out***.println("Short wrapper class: " + value6);

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

String strNumber4 = "12345";

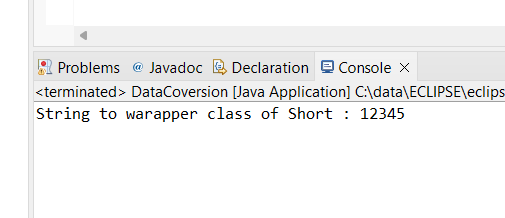
**short** value7 = Short.*valueOf*(strNumber4);

System.***out***.println("String to warapper class of Short : " + value7);

}

}

Output:



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

Program:

**package** org.example.Assignment2;

**public** **class** DataCoversion {

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

**char** char1 = '1';

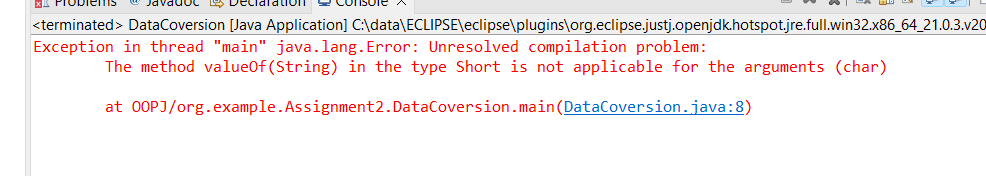
**short** value8 = Short.*valueOf*(char1);

System.***out***.println("Char to Short : " + value8);

}

}

Output:



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

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

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

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

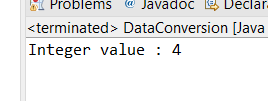
**int** value = Integer.***BYTES***;

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

}

}

Output:



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

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**int** Value1 = Integer.***MAX\_VALUE***;

**int** Value2 = Integer.***MIN\_VALUE***;

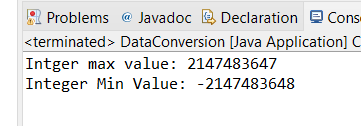
System.***out***.println("Intger max value: " + Value1);

System.***out***.println("Integer Min Value: " + Value2);

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**int** Number = 123569;

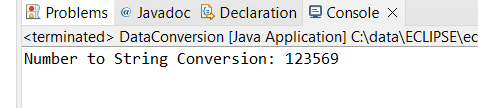
String Value3 = Integer.*toString*(Number);

System.***out***.println("Number to String Conversion: " + Value3);

}

}

Output:



**e.** Declare a method-local variable strNumber of type String with some value and convert it to an int value using the parseInt method. (Hint: Use Integer.parseInt(String)).

**Program:**

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

String strNumber = "12345";

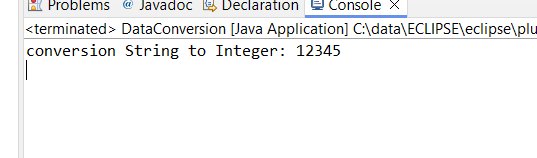
**int** value4 = Integer.*parseInt*(strNumber);

System.***out***.println("conversion String to Integer: " + value4);

}

}

Output:



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

**Program:**

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

String strNumber1 = "ABCDEF";

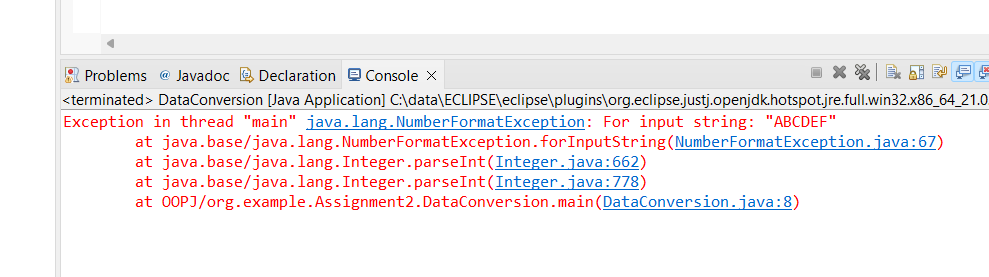
**int** value5 = Integer.*parseInt*(strNumber1);

System.***out***.println("conversion String to Integer: " + value5);

}

}

Output:



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

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**int** number1 = 4567;

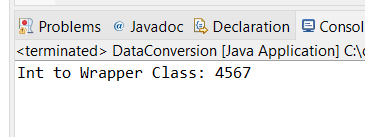
**int** value6 = Integer.*valueOf*(number1);

System.***out***.println("Int to Wrapper Class: " + value6);

}

}

Output:



**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)).

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

String strNumber3 = "234543";

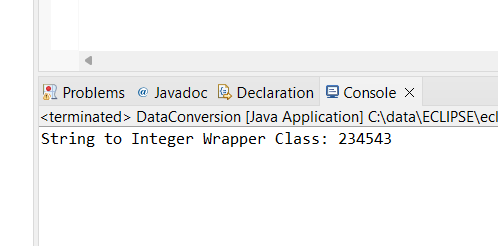
**int** value7 = Integer.*valueOf*(strNumber3);

System.***out***.println("String to Integer Wrapper Class: " + value7);

}

}

Output:



**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)).

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**int** value8 = 10;

**int** value9 = 20;

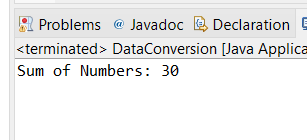
**int** sum = Integer.*sum*(value8, value9);

System.***out***.println("Sum of Numbers: " + sum);

}

}

Output:



**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)).

**Program:**

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**int** num1 = 10;

**int** num2 = 20;

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

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

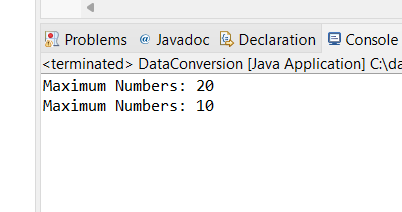
System.***out***.println("Maximum Numbers: " + max);

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

}

}

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)).

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**int** num3 = 7;

String Binary = Integer.*toBinaryString*(num3);

String Octal = Integer.*toOctalString*(num3);

String Hexa = Integer.*toHexString*(num3);

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

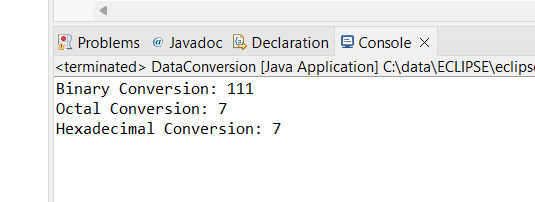
System.***out***.println("Octal Conversion: " + Octal);

System.***out***.println("Hexadecimal Conversion: " + Hexa);

}

}

Output:



#### ****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).

**program:**

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

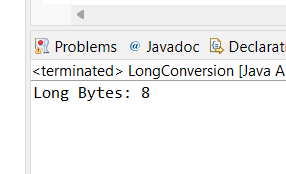
**long** num1=Long.***BYTES***;

System.***out***.println("Long Bytes: "+num1);

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**long** num2 = 23453267;

**long** num3 = 567463589;

**long** max = Long.*max*(num2, num3);

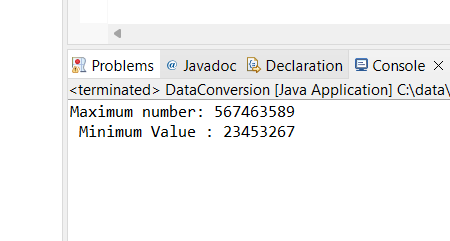
**long** min = Long.*min*(num2, num3);

System.***out***.println("Maximum number: " + max + "\n Minimum Value : " + min);

}

}

Output:



**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)).

Program

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**long** num4 = 345678;

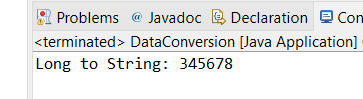
String str = Long.*toString*(num4);

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

}

}

Output:



**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)).

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

String strNumber1 = "56784";

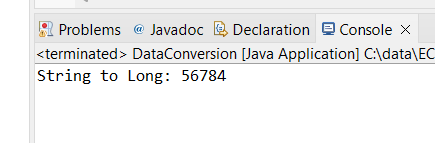
**long** value = Long.*parseLong*(strNumber1);

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

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

String strType = "Ab12cd34";

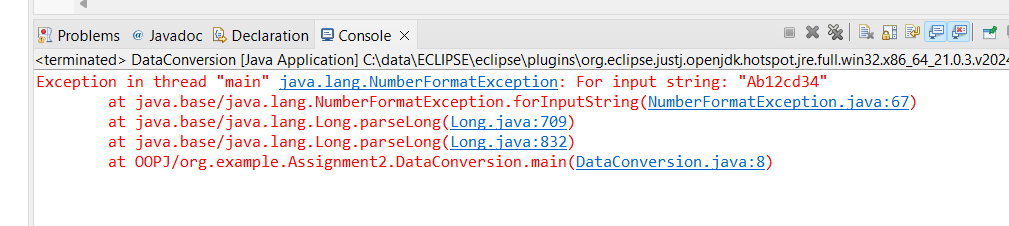
**long** value2 = Long.*parseLong*(strType);

System.***out***.println("String Value into Long: " + value2);

}

}

Output:



**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)).

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**long** num5 = 567890;

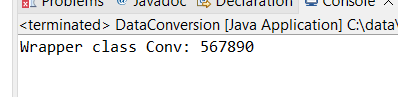
**long** value3 = Long.*valueOf*(num5);

System.***out***.println("Wrapper class Conv: " + value3);

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

String strNumber = "5674325364";

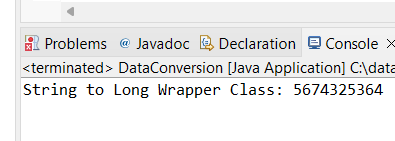
**long** value4 = Long.*valueOf*(strNumber);

System.***out***.println("String to Long Wrapper Class: " + value4);

}

}

Output:



**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)).

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**long** var1 = 1123;

**long** var2 = 9845;

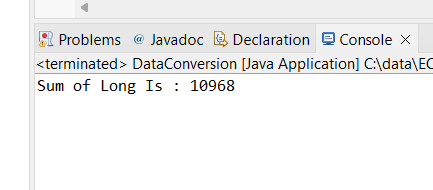
**long** sum = Long.*sum*(var1, var2);

System.***out***.println("Sum of Long Is : " + sum);

}

}

Output:



**j.** Declare two long variables with values 1122 and 5566, and find the minimum and maximum values using the Long class. (Hint: Use Long.min(long, long) and Long.max(long, long)).

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**long** var3 = 1122;

**long** var4 = 5566;

**long** max1 = Long.*max*(var3, var4);

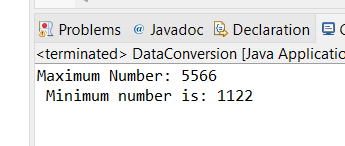
**long** min1 = Long.*min*(var3, var4);

System.***out***.println("Maximum Number: " + max1 + "\n Minimum number is: " + min1);

}

}

Output:



**k.** Declare a long variable with the value 7. Convert it to binary, octal, and hexadecimal strings using methods from the Long class. (Hint: Use Long.toBinaryString(long), Long.toOctalString(long), and Long.toHexString(long)).

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**long** var5 = 7;

System.***out***.println("Long to Binary: " + Long.*toBinaryString*(var5));

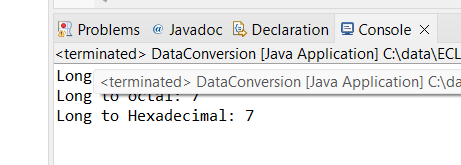
System.***out***.println("Long to octal: " + Long.*toOctalString*(var5));

System.***out***.println("Long to Hexadecimal: " + Long.*toHexString*(var5));

}

}

Output:



**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).

**package** org.example.Assignment2;

**public** **class** DataConversion {

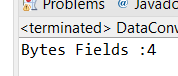
**public** **static** **void** main(String[] args) {

System.***out***.println("Bytes Fields :" + Float.***BYTES***);

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

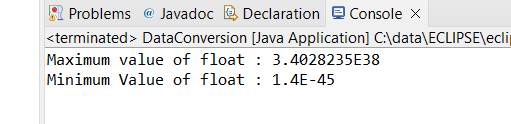
System.***out***.println("Maximum value of float : "+Float.***MAX\_VALUE***);

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

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

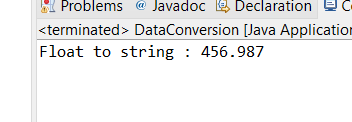
**float** num = 456.987f;

System.***out***.println("Float to string : " + Float.*toString*(num));

}

}

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)).

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

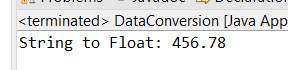
String strNumber = "456.78";

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

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

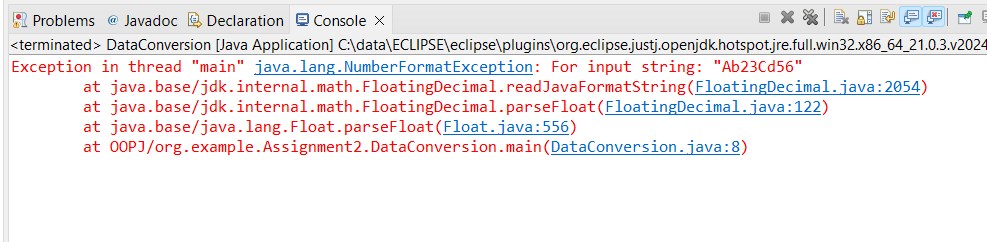
String strNumber2 = "Ab23Cd56";

System.***out***.println("String to Float with Charcter: " + Float.*parseFloat*(strNumber2));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

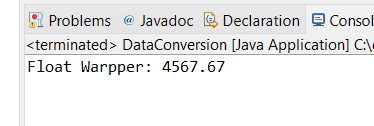
**float** num4 = 4567.67f;

System.***out***.println("Float Warpper: " + Float.*valueOf*(num4));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

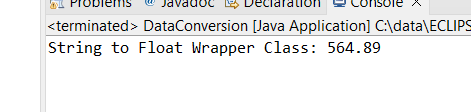
String strNumber3 = "564.890";

System.***out***.println("String to Float Wrapper Class: " + Float.*valueOf*(strNumber3));

}

}

Output:



**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)).

Program:

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

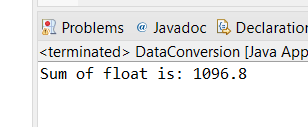
**float** num5 = 112.3f, num6 = 984.5f;

System.***out***.println("Sum of float is: " + Float.*sum*(num5, num6));

}

}

Output:



**j.** Declare two float variables with values 112.2 and 556.6, and find the minimum and maximum values using the Float class. (Hint: Use Float.min(float, float) and Float.max(float, float)).

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

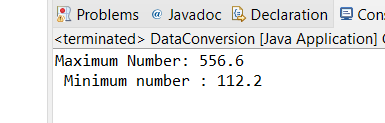
**float** num7=112.2f,num8=556.6f;

System.***out***.println("Maximum Number: "+ Float.*max*(num7,num8) + "\n Minimum number : "+ Float.*min*(num7, num8));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**float** num9=-25.0f;

System.***out***.println("Square root is: "+Math.*sqrt*(num9));

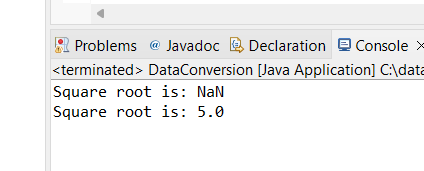
**float** num10 =25.0f;

System.***out***.println("Square root is: "+Math.*sqrt*(num10));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**float** num10 = 0.0f, num11 = 0.0f;

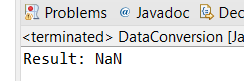
**float** div = num10 / num11;

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

}

}

Output:



**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).

**package** org.example.Assignment2;

**public** **class** DataConversion {

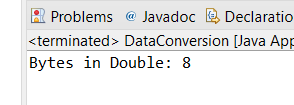
**public** **static** **void** main(String[] args) {

System.***out***.println("Bytes in Double: " + Double.***BYTES***);

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

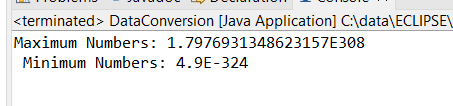
**public** **static** **void** main(String[] args) {

System.***out***.println("Maximum Numbers: " + Double.***MAX\_VALUE*** + "\n Minimum Numbers: " + Double.***MIN\_VALUE***);

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

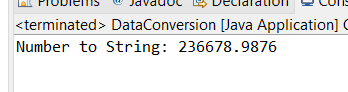
**double** Number=236678.9876;

System.***out***.println("Number to String: "+Double.*toString*(Number));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

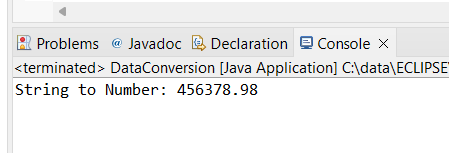
String strNumber = "456378.98";

System.***out***.println("String to Number: " + Double.*parseDouble*(strNumber));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

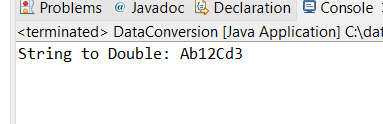
String strNumber2="Ab12Cd3";

System.***out***.println("String to Double: "+ strNumber2);

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

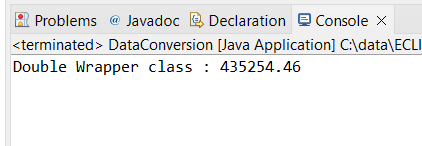
**double** number2=435254.46;

System.***out***.println("Double Wrapper class : "+ Double.*valueOf*(number2));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

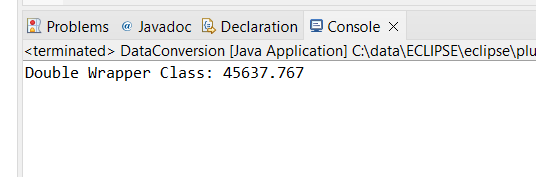
String strNumber3="45637.767";

System.***out***.println("Double Wrapper Class: "+Double.*valueOf*(strNumber3));

}

}

Output:



**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)).

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

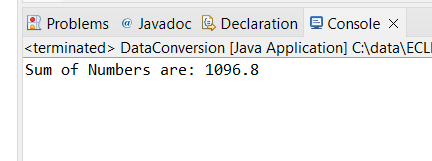
**double** num1 = 112.3, num2 = 984.5;

System.***out***.println("Sum of Numbers are: " + Double.*sum*(num1, num2));

}

}

**Output:**

****

**j.** Declare two double variables with values 112.2 and 556.6, and find the minimum and maximum values using the Double class. (Hint: Use Double.min(double, double) and Double.max(double, double)).

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

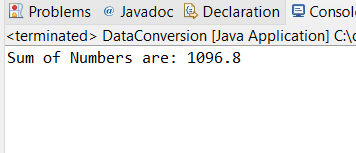
**double** num3 = 112.2, num4 = 556.6;

System.***out***.println("maximum Number: " + Double.*max*(num3, num4) + "\nMinimum Value: " + Double.*min*(num3, num4));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

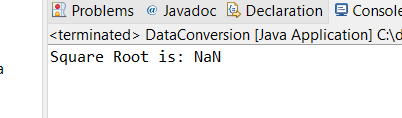
**double** num5 = -25.0;

System.***out***.println("Square Root is: " + Math.*sqrt*(num5));

}

}

Output:



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

**package** org.example.Assignment2;

**public** **class** DataConversion {

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

**double** num6=0.0,num7=0.0;

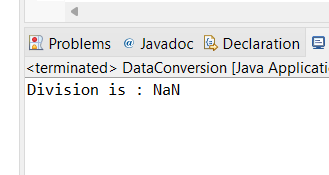
**double** div=num6/num7;

System.***out***.println("Division is : "+div);

}

}

Output:



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

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

#### ****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).

#### ****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).