Sandeep Sir’s Assignment

# Question 1

**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**

public class Q1{

    public static void main(String[] args) {

        boolean b = true;

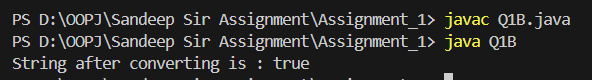
        String s = Boolean.toString(b);

        System.out.println(s)003B

    }

}

**Output :-**

****

**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 Q1C {

    public static void main(String[] args) {

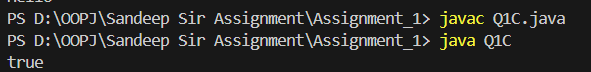
        String strStatus = "true";

        boolean b = Boolean.parseBoolean(strStatus);

        System.out.println(b);

    }

}

****

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

public class Q1D {

    public static void main(String[] args) {

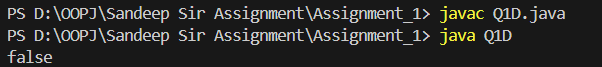
        String strStatus = "1";

        boolean b = Boolean.parseBoolean(strStatus);

        System.out.println(b);

    }

}



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

public class Q1E {

    public static void main(String[] args) {

        boolean status = true;

        Boolean b = Boolean.valueOf(status);

        System.out.println(b);

    }

}



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

public class Q1F {

    public static void main(String[] args) {

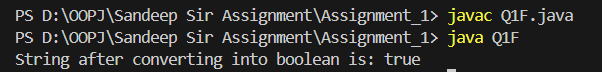
        String strStatus = "true";

        Boolean b = Boolean.valueOf(strStatus);

        System.out.println("String after converting into boolean is: "+ b);

    }

}

****

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

# Question 2 -

**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 Q2B {

    public static void main(String[] args) {

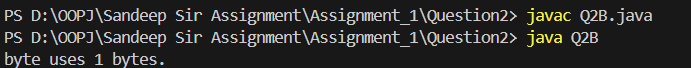
        int byteused = Byte.BYTES;

        System.out.println("byte uses "+ byteused + " 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) {

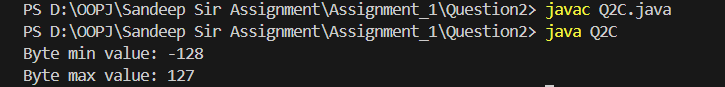
        System.out.println("Byte min value: "+ Byte.MIN\_VALUE);

        System.out.println("Byte max value: "+ Byte.MAX\_VALUE);

    }

}

**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)).**

public class Q2D {

    public static void main(String[] args) {

        byte number = 2;

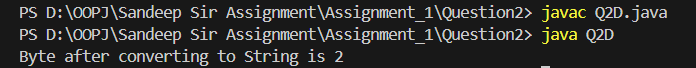
        String s = Byte.toString(number);

        System.out.println("Byte after converting to String is "+ s );

    }

}

**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 = "12";

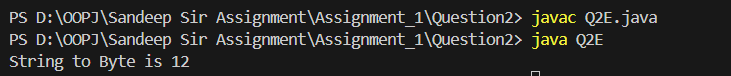
        byte b = Byte.parseByte(strNumber);

        System.out.println("String to Byte is "+ b);

    }

}

**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) {

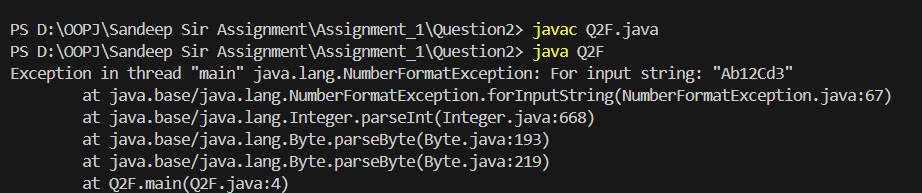
        String strNumber = "Ab12Cd3";

        byte b = Byte.parseByte(strNumber);

        System.out.println("String to byte : ");

    }

}

****

**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) {

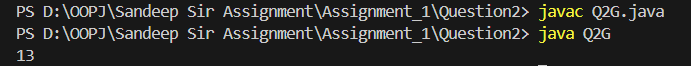
        byte number = 13;

        System.out.println(Byte.valueOf(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 = "127";

        System.out.println(Byte.valueOf(strNumber));

    }

}

**Output -**



**I. 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) {

        byte b = 125;

        short s = b;

        int i = b;

        float f = b;

        long l = b;

        double d = b;

        System.out.println("In short: " + s);

        System.out.println("In int: " + i);

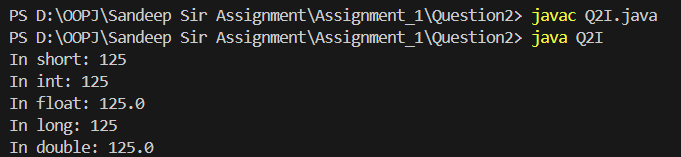
        System.out.println("In float: " + f);

        System.out.println("In long: " + l);

        System.out.println("In double: " + d);

    }

}



# Question 3 - Working with java.lang.Short

**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 Q3A {

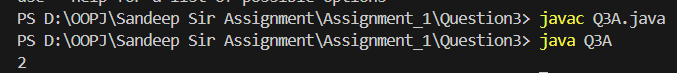
    public static void main(String[] args) {

        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 Q3B {

    public static void main(String[] args) {

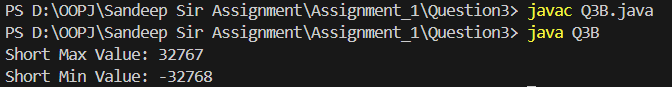
        System.out.println("Short Max Value: "+Short.MAX\_VALUE);

        System.out.println("Short Min Value: "+Short.MIN\_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 = 12456;

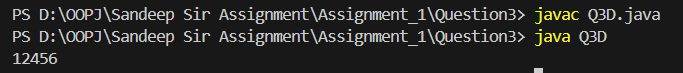
        String s = Short.toString(number);

        System.out.println(s);

    }

}

**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 = "14545";

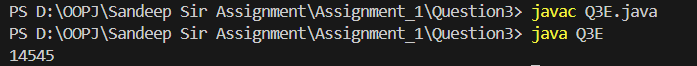
        short s = Short.parseShort(strNumber);

        System.out.println(s);

    }

}

**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).**

public class Q3F {

    public static void main(String[] args) {

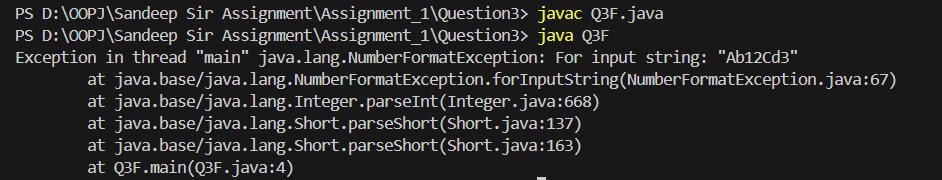
        String strNumber = "Ab12Cd3" ;

        System.out.println(Short.parseShort(strNumber));

    }

}

**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 {

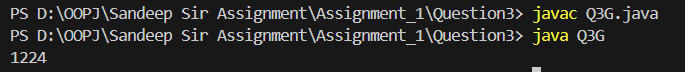
    public static void main(String[] args) {

        short number = 1224;

        System.out.println(Short.valueOf(number));

    }

}

**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) {

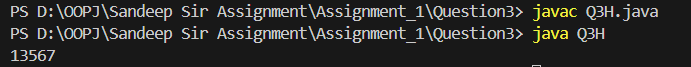
        String strNumber = "13567";

        System.out.println(Short.valueOf(strNumber));

    }

}

**Output –**

****

**I. 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 s = 125;

        byte b = (byte)s;

        int i = b;

        float f = b;

        long l = b;

        double d = b;

        System.out.println("In byte: " + b);

        System.out.println("In int: " + i);

        System.out.println("In float: " + f);

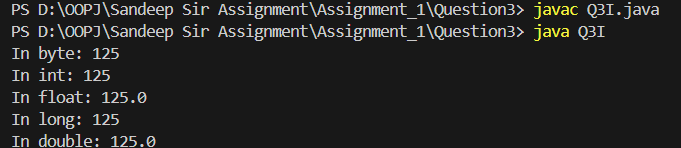
        System.out.println("In long: " + l);

        System.out.println("In double: " + d);

    }

}

**Output –**

****

# Question 4 - Working with java.lang.Integer

**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 {

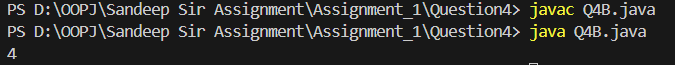
    public static void main(String[] args) {

        System.out.println(Integer.BYTES);

    }

}

**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).**

public class Q4C{

    public static void main(String[] args) {

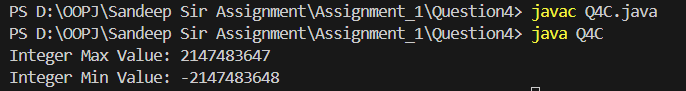
        System.out.println("Integer Max Value: "+Integer.MAX\_VALUE);

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

    }

}

**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)).**

public class Q4D {

    public static void main(String[] args) {

        int number = 124564565;

        String s = Integer.toString(number);

        System.out.println(s);

    }

}

**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)).**

public class Q4E {

    public static void main(String[] args) {

        String strNumber = "14545";

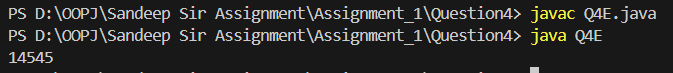
        int s = Integer.parseInt(strNumber);

        System.out.println(s);

    }

}

**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).**

public class Q4F {

    public static void main(String[] args) {

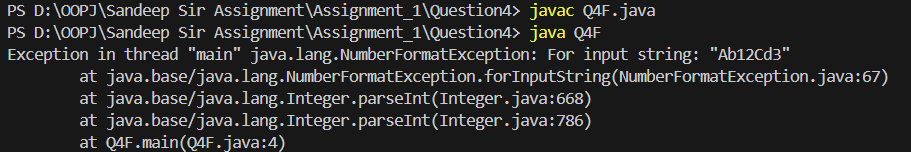
        String strNumber = "Ab12Cd3" ;

        System.out.println(Integer.parseInt(strNumber));

    }

}

**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)).**

public class Q4G {

    public static void main(String[] args) {

        int number = 1224;

        System.out.println(Integer.valueOf(number));

    }

}

**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)).**

public class Q4H {

    public static void main(String[] args) {

        String strNumber = "13567";

        System.out.println(Integer.valueOf(strNumber));

    }

}

**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)).**

public class Q4I {

    public static void main(String[] args) {

        int a = 10;

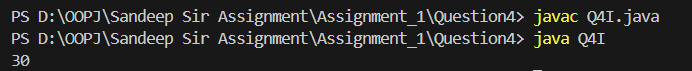
        int b = 20;

        System.out.println(Integer.sum(a, b));

    }

}

**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)).**

public class Q4J {

    public static void main(String[] args) {

        int a = 10;

        int b = 20;

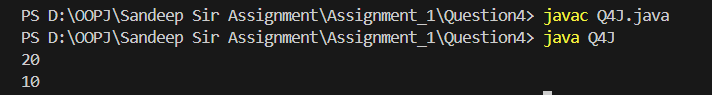
        System.out.println(Integer.max(a, b));

        System.out.println(Integer.min(a, b));

    }

}

**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)).**

public class Q4K {

    public static void main(String[] args) {

        int i = 7;

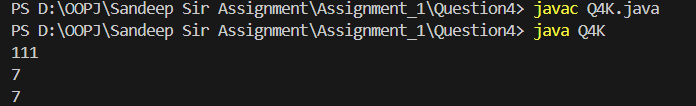
        System.out.println(Integer.toBinaryString(i));

        System.out.println(Integer.toOctalString(i));

        System.out.println(Integer.toHexString(i));

    }

}

**Output - **

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

public class Q4L {

    public static void main(String[] args) {

        int s = 12546;

        byte b = (byte)s;

        int i = b;

        float f = b;

        long l = b;

        double d = b;

        System.out.println("In byte: " + b);

        System.out.println("In int: " + i);

        System.out.println("In float: " + f);

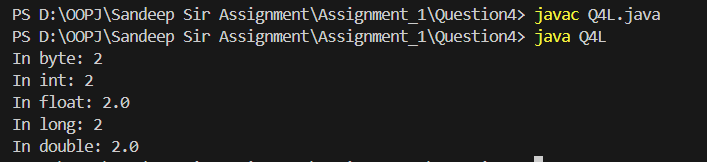
        System.out.println("In long: " + l);

        System.out.println("In double: " + d);

    }

}

**Output –**

****

# Question 5

**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 {

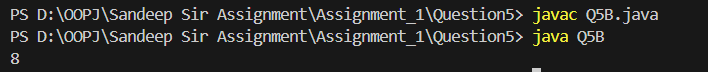
    public static void main(String[] args) {

        System.out.println(Long.BYTES);

    }

}

**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).**

public class Q5C{

    public static void main(String[] args) {

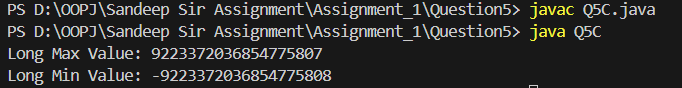
        System.out.println("Long Max Value: "+Long.MAX\_VALUE);

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

    }

}

**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)).**

public class Q5D {

    public static void main(String[] args) {

        int number = 124564565;

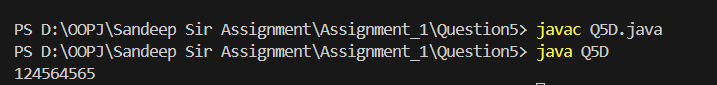
        String s = Long.toString(number);

        System.out.println(s);

    }

}

**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)).**

public class Q5E {

    public static void main(String[] args) {

        String strNumber = "14545";

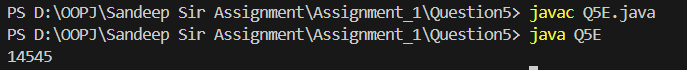
        long s = Long.parseLong(strNumber);

        System.out.println(s);

    }

}

**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).**

public class Q5F {

    public static void main(String[] args) {

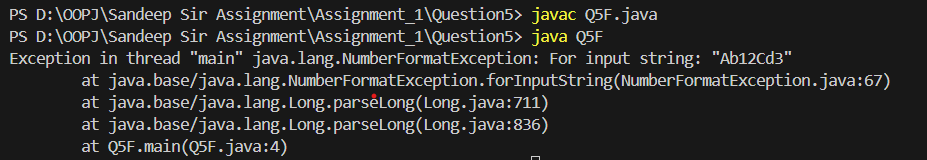
        String strNumber = "Ab12Cd3" ;

        System.out.println(Long.parseLong(strNumber));

    }

}

**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)).**

public class Q5G {

    public static void main(String[] args) {

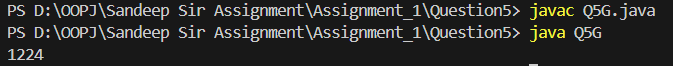
        long number = 1224;

        System.out.println(Long.valueOf(number));

    }

}

**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)).**

public class Q5H {

    public static void main(String[] args) {

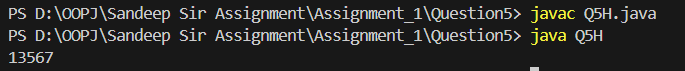
        String strNumber = "13567";

        System.out.println(Long.valueOf(strNumber));

    }

}

**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)).**

public class Q5I {

    public static void main(String[] args) {

        long a = 1123L;

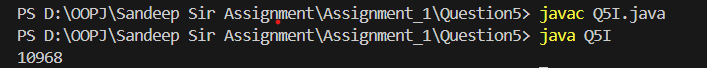
        long b = 9845L;

        System.out.println(Long.sum(a, b));

    }

}

**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)).**

public class Q5J {

    public static void main(String[] args) {

        long a = 1166L;

        long b = 5566L;

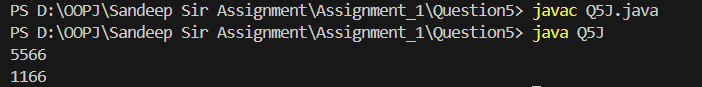
        System.out.println(Long.max(a, b));

        System.out.println(Long.min(a, b));

    }

}

**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)).**

public class Q5K {

    public static void main(String[] args) {

        long i = 7L;

        System.out.println(Long.toBinaryString(i));

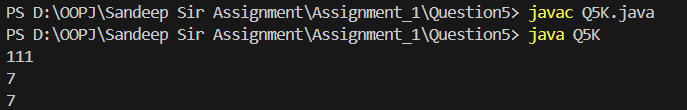
        System.out.println(Long.toOctalString(i));

        System.out.println(Long.toHexString(i));

    }

}

**Output –**

****

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

public class Q5L {

    public static void main(String[] args) {

        long l = 1254667L;

        byte b = (byte)l;

        int i = b;

        float f = b;

        short s = b;

        double d = b;

        System.out.println("In byte: " + b);

        System.out.println("In short: " + s);

        System.out.println("In int: " + i);

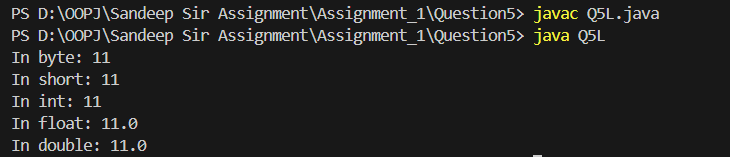
        System.out.println("In float: " + f);

        System.out.println("In double: " + d);

    }

}

**Output –**

****

# Question 6

**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 {

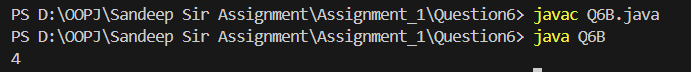
    public static void main(String[] args) {

        System.out.println(Float.BYTES);

    }

}

**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).**

public class Q6C{

    public static void main(String[] args) {

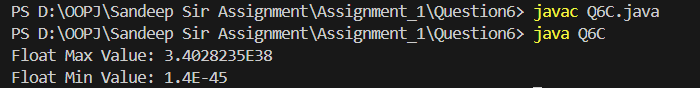
        System.out.println("Float Max Value: "+Float.MAX\_VALUE);

        System.out.println("Float Min Value: "+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)).**

public class Q6D {

    public static void main(String[] args) {

        float number = 2.5f;

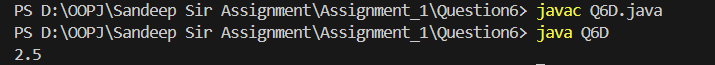
        String s = Float.toString(number);

        System.out.println(s);

    }

}

**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 = "145.0";

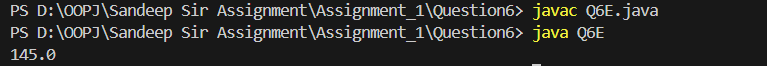
        float s = Float.parseFloat(strNumber);

        System.out.println(s);

    }

}

**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).**

public class Q6F {

    public static void main(String[] args) {

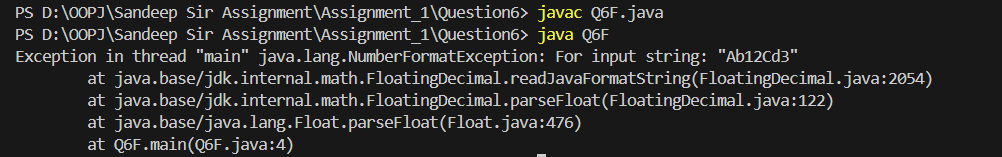
        String strNumber = "Ab12Cd3" ;

        System.out.println(Float.parseFloat(strNumber));

    }

}

**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)).**

public class Q6G {

    public static void main(String[] args) {

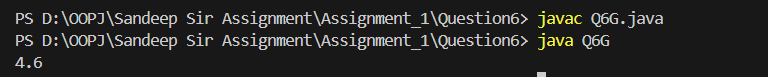
        float number = 4.6f;

        System.out.println(Float.valueOf(number));

    }

}

**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)).**

public class Q6H {

    public static void main(String[] args) {

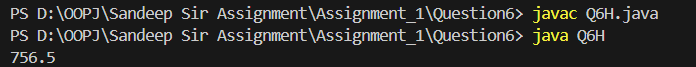
        String strNumber = "756.5";

        System.out.println(Float.valueOf(strNumber));

    }

}

**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)).**

public class Q6I {

    public static void main(String[] args) {

        float a = 112.3f;

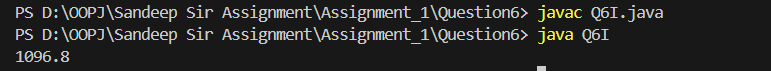
        float b = 984.5f;

        System.out.println(Float.sum(a, b));

    }

}

**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)).**

public class Q6J {

    public static void main(String[] args) {

        float a = 112.2f;

        float b = 556.6f;

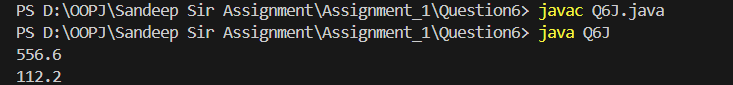
        System.out.println(Float.max(a, b));

        System.out.println(Float.min(a, b));

    }

}

**Output –**

****

**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) {

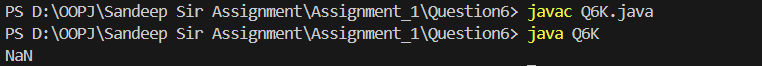
        float f = -25.0f;

        System.out.println(Math.sqrt(f));

    }

}

**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).**

public class Q6L {

    public static void main(String[] args) {

        float f1 = 0.0f;

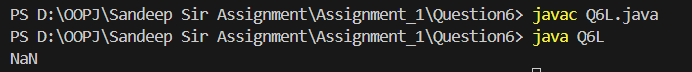
        float f2 = 0.0f;

        System.out.println(f1/f2);

    }

}

**Output –**

****

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

public class Q6M {

    public static void main(String[] args) {

        float f = 2.5f;

        byte b = (byte)f;

        short s = (short)f;

        int i = (int)f;

        long l = (long) f;

        double d = f;

        System.out.println("In byte: " + b);

        System.out.println("In short: " + s);

        System.out.println("In int: " + i);

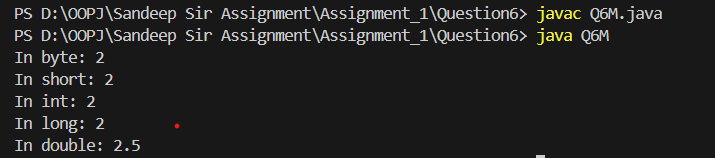
        System.out.println("In long: " + l);

        System.out.println("In double: " + d);

    }

}

**Output –**

****

# Question 7

**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);

    }

}

**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).**

public class Q7C{

    public static void main(String[] args) {

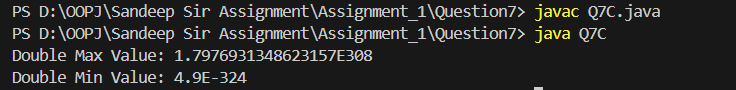
        System.out.println("Double Max Value: " +Double.MAX\_VALUE);

        System.out.println("Double Min Value: " +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)).**

public class Q7D {

    public static void main(String[] args) {

        Double number = 2.5;

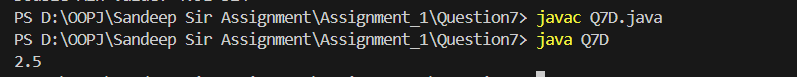
        String s = Double.toString(number);

        System.out.println(s);

    }

}

**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)).**

public class Q7E {

    public static void main(String[] args) {

        String strNumber = "145.0";

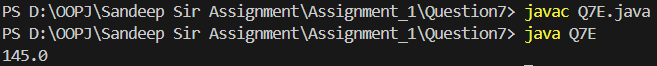
        double s = Double.parseDouble(strNumber);

        System.out.println(s);

    }

}

**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).**

public class Q7F {

    public static void main(String[] args) {

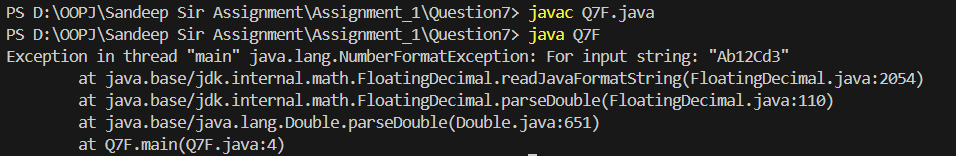
        String strNumber = "Ab12Cd3" ;

        System.out.println(Double.parseDouble(strNumber));

    }

}

**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)).**

public class Q7G {

    public static void main(String[] args) {

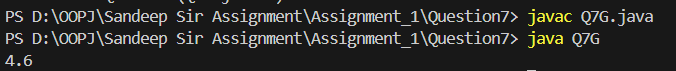
        double number = 4.6;

        System.out.println(Double.valueOf(number));

    }

}

**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)).**

public class Q7H {

    public static void main(String[] args) {

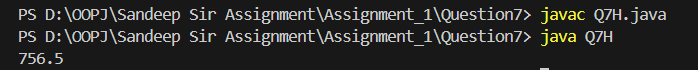
        String strNumber = "756.5";

        System.out.println(Double.valueOf(strNumber));

    }

}

**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)).**

public class Q7I {

    public static void main(String[] args) {

        double a = 112.3;

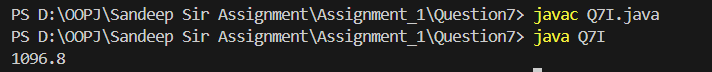
        double b = 984.5;

        System.out.println(Double.sum(a, b));

    }

}

**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)).**

public class Q7J {

    public static void main(String[] args) {

        double a = 112.2;

        double b = 556.6;

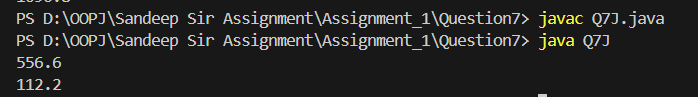
        System.out.println(Double.max(a, b));

        System.out.println(Double.min(a, b));

    }

}

**Outuput –**

****

**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) {

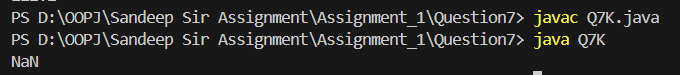
        double d = -25.0;

        System.out.println(Math.sqrt(d));

    }

}

**Output –**

****

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

public class Q7L {

    public static void main(String[] args) {

        double d1 = 0.0;

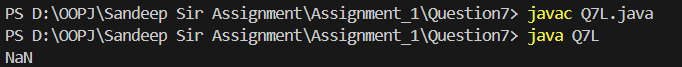
        double d2 = 0.0;

        System.out.println(d1/d2);

    }

}

**Output –**

****

**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 d = 2.5;

        byte b = (byte)d;

        short s = (short)d;

        int i = (int)d;

        float f = (float)d;

        long l = (long)d;

        System.out.println("In byte: " + b);

        System.out.println("In short: " + s);

        System.out.println("In int: " + i);

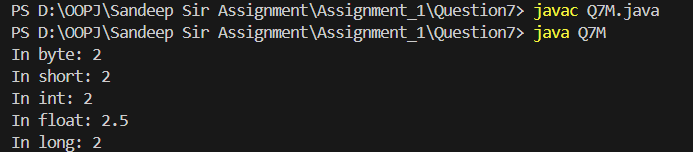
        System.out.println("In float: " + f);

        System.out.println("In long: " + l);

    }

}

**Output –**

****

# Question 8

public class Q8 {

    public static void main(String[] args) {

        boolean bl = true;

        byte b = 45;

        short s = 466;

        int i = 235;

        float f = 25.76f;

        long l = 4556L;

        double d = 45.77;

        // Boolean

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

        System.out.println("Value of above: " + String.valueOf(Boolean.toString(bl)));

        // Byte

        System.out.println("Byte to String: " + Byte.toString(b));

        System.out.println("Value of above: " + String.valueOf(Byte.toString(b)));

        // Short

        System.out.println("Short to String: " + Short.toString(s));

        System.out.println("Value of above: " + String.valueOf(Short.toString(s)));

        // Integer

        System.out.println("Integer to String: " + Integer.toString(i));

        System.out.println("Value of above: " + String.valueOf(Integer.toString(i)));

        // Float

        System.out.println("Float to String: " + Float.toString(f));

        System.out.println("Value of above: " + String.valueOf(Float.toString(f)));

        // Long

        System.out.println("Long to String: " + Long.toString(l));

        System.out.println("Value of above: " + String.valueOf(Long.toString(l)));

        // Double

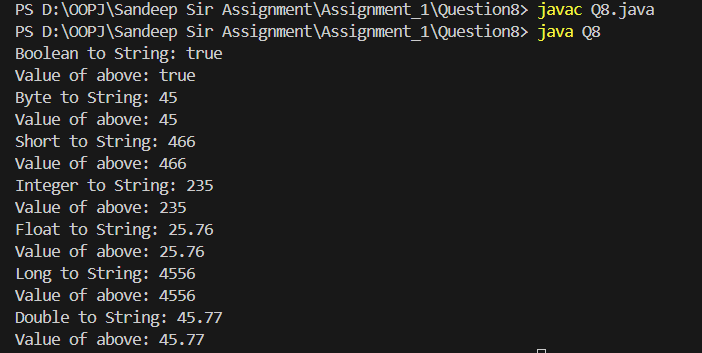
        System.out.println("Double to String: " + Double.toString(d));

        System.out.println("Value of above: " + String.valueOf(Double.toString(d)));

    }

}

**Output –**



# Question 10

## Arithmetic Operation with Command Line

**10. 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).**

import java.util.Scanner;

public class Q10 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter first number: ");

        int a = sc.nextInt();

        System.out.println("Enter second number: ");

        int b = sc.nextInt();

        System.out.println("(+, -, \*, /)Choose the operator: ");

        char op = sc.next().charAt(0);

        int result=0;

        if(op == '+'){

            result = a+b;

        }

        else if(op == '-'){

            result = a-b;

        }

        else if(op == '\*'){

            result = a\*b;

        }

        else if(op == '/'){

            if(b == 0){

                result = 0;

            }else

            {

                result = a/b;

            }

        }

        System.out.println(result);

    }

}

