**Name: Yash Bandu Dhole**

**Center: Juhu.**

**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.

**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) ).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        boolean status = true;

        String str = Boolean.toString(status);

        System.out.println("Convertedt string is: " + str);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Convertedt string is: true

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        //boolean status = true;

        String strStatus = "true";

        boolean status = Boolean.parseBoolean(strStatus);

        System.out.println("String Converted To Bool is: " + status );

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> String Converted To Bool is: true

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        //boolean status = true;

        String strStatus = "1";

        boolean status = Boolean.parseBoolean(strStatus);

        System.out.println("String Converted To Bool is: " + status );

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> String Converted To Bool is: false

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        boolean status = true;

Boolean statusWrapper = Boolean.valueOf(status)

        System.out.println("The status as a Boolean object: " + statusWrapper);

    }

}

o/p: PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> status as a Boolean object: true

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        String strStatus = "true";

        Boolean statusWrapper = Boolean.valueOf(strStatus);

        System.out.println("The status as a Boolean object: " + statusWrapper);

    }

}

o/p: PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> status as a Boolean object: true

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        int number = 5;

        boolean intToBool = (number != 0);

        System.out.println("Int to boolean: " + intToBool);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Int to boolean: true

#### ****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).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        int bytesUsed = Byte.BYTES;

        System.out.println("Number of bytes used to represent a byte value: " + bytesUsed);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Number of bytes used to represent a byte value: 1

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        byte minValue = Byte.MIN\_VALUE;

        byte maxValue = Byte.MAX\_VALUE;

        System.out.println("Minimum value of byte: " + minValue);

        System.out.println("Maximum value of byte: " + maxValue);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Minimum value of byte: -128

Maximum value of byte: 127

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        byte number =12;

        String str = Byte.toString(number);

        System.out.println("converted byte to str:" + str);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> converted byte to str:12

**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)).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       String strNumber = "Yash";

       byte bt = Byte.parseByte(strNumber);

    }

}

o/p:

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

at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)

at java.lang.Integer.parseInt(Integer.java:580)

at java.lang.Byte.parseByte(Byte.java:149)

at java.lang.Byte.parseByte(Byte.java:175)

at methodLocal.main(methodLocal.java:5)

/\* Exception occurs \*/

**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).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       String strNumber = "Ab12Cd3";

       byte bt = Byte.parseByte(strNumber);

    }

}

o/p: Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"

**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)).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       byte number = 12;

       Byte bt = Byte.valueOf(number);

       System.out.println(" Converting byte to its Wrapper: " +bt);

    }

}

**o/p:**

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Converting byte to its Wrapper: 12

**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)).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       String strNumber = "12";

       Byte bt = Byte.valueOf(strNumber);

       System.out.println("byte string to Wrapper byte: " + bt);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> byte string to Wrapper byte: 12

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

      byte bt = 11;

      long lg = Byte.valueOf(bt);

      System.out.println("converting to "+lg);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> converting to 11

#### ****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).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        int bytesUsed = Short.BYTES;

        System.out.println(" Number of bytes used to represent a short value: " + bytesUsed);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Number of bytes used to represent a short value: 2

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

        System.out.println(" Range of Short from : " + Short.MIN\_VALUE);

        System.out.println(" Range of Short To : " + Short.MAX\_VALUE);

    }

}

o/p:

Range of Short from : -32768

Range of Short To : 32767

**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)).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       short number =123;

       String str = Short.toString(number);

       System.out.println("short to string: "+str );

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> short to string: 123

**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)).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       String strNumber = "123";

       short st = Short.parseShort(strNumber);

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

    }

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> String to Short: 123

**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).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       String strNumber = "Ab12Cd3";

       short st = Short.parseShort(strNumber);

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

    }

}

o/p:

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

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

    public static void main(String[] args) {

       short number =123;

       Short st = Short.valueOf(number);

       System.out.println("Short to Wrapper: "+st);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Short to Wrapper: 12

**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 methodLocal {

    public static void main(String[] args) {

       String strNumber = "123";

       Short st = Short.valueOf(strNumber);

       System.out.println("String short to Wrapper Short: "+ st);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> String short to Wrapper Short: 123

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

Code:

public class methodLocal {

    public static void main(String[] args) {

        short a = 321;

        long var = Short.valueOf(a);

       System.out.println("short to long: "+ var);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> short to long: 321

#### ****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).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       System.out.println("Bytes for Integer: "+ Integer.BYTES);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Bytes for Integer: 4

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

       System.out.println("Range of Integer starts: "+ Integer.MIN\_VALUE);

       System.out.println("Range of Integer stops: "+ Integer.MAX\_VALUE);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Range of Integer starts: -2147483648

Range of Integer stops: 2147483647

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

      int number = 123;

      String str =Integer.toString(number);

      System.out.println("integer to string: " + str);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> integer to string: 123

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

public class methodLocal {

    public static void main(String[] args) {

      String strNumber = "123";

      int a = Integer.parseInt(strNumber);

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

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> String to Int: 123

**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).  
Code:

public class methodLocal {

    public static void main(String[] args) {

      String strNumber = "Ab12Cd3";

      int a = Integer.parseInt(strNumber);

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

    }

}

o/p:

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

**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)).  
Code:

public class methodLocal {

    public static void main(String[] args) {

      int number =123;

      Integer wrapNum = Integer.valueOf(number);

      System.out.println("Wrapping number: " + wrapNum);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Wrapping number: 123

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

      String strNumber = "123";

      Integer a = Integer.parseInt(strNumber);

      System.out.println("Wrapping number: " + a);

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Wrapping number: 123

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

**Code:**

public class methodLocal {

    public static void main(String[] args) {

      int a=10;

      int b= 20;

      System.out.println("addition of 2 Int " +Integer.sum(a, b));

    }

}

**o/p:**

**PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> addition of 2 Int 30**

**j.** Declare two integer variables with values 10 and 20, and find the minimum and maximum values using the Integer class. (Hint: Use Integer.min(int, int) and Integer.max(int, int)).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

      int a=10;

      int b= 20;

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

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

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> 10

20

**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)).

**Code:**

public class methodLocal {

    public static void main(String[] args) {

      int a=7;

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

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

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

    }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> 7

7

**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).

**Code:**

public class methodLocal {

  public static void main(String[] args) {

     System.out.println("Bytes for Long: "+ Long.BYTES);

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Bytes for Long: 8

**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).

**Code:**

public class methodLocal {

  public static void main(String[] args) {

     System.out.println("range of long: "+ Long.MIN\_VALUE);

     System.out.println("range of long till: "+ Long.MAX\_VALUE);

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> range of long: -9223372036854775808

range of long till: 9223372036854775807

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

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  long n=741;

  String str = Long.toString(n);

  System.out.println(str);

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> 741

**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)).

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  String strNumber = "852";

  long lg = Long.parseLong(strNumber);

  System.out.println(lg);

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> 852

**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).

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  String strNumber = "Ab12Cd3";

  long lg = Long.parseLong(strNumber);

  System.out.println(lg);

  }

}

**o/p:**

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"

**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)).

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  long number = 564;

  Long lg =Long.valueOf(number);

  System.out.println(lg);

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> 564

**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)).

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  String strNumber ="10010";

  Long lg = Long.valueOf(strNumber);

  System.out.println(lg);

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> 10010

**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)).

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  long a = 1123;

  long b = 9845;

  System.out.println(Long.sum(a, b));

  }

}

o/p:

**PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> 10968**

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

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  long a = 1122;

  long b = 5566;

  System.out.println(Long.min(a, b));

  System.out.println(Long.max(a, b));

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes>1122

5566

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

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  long a = 7;

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

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

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

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> 111

7

7

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

**Code:**

public class methodLocal {

  public static void main(String[] args) {

  long a = 7;

  int v1 = Long.valueOf(a);

  }

}

o/p:

PS D:\Yash\C-DAC\MODULE 2 core-java\Codes> Error: cannot convert from Long to int**.**