

# Sandeep Sir: OOPJ Assignment 2

## 1. Working with java.lang.Boolean

a. Explore the Java API documentation for java.lang.Boolean 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) )

```
class Ques1{
    public static void main(String args[]){
        boolean status = true;
        String str = Boolean.toString(status);
        System.out.println(str);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques1.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques1
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)).

```
class Ques1{
    public static void main(String args[]){
        String strStatus = "true";
        boolean status = Boolean.parseBoolean(strStatus);
    }
}
```

```

        System.out.println(status);
    }
}

```

```

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques1.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques1
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").

→ Parses the string argument as a boolean. The boolean returned represents the value true if the string argument is not null and is equal, ignoring case, to the string "true"

```

class Ques1{
    public static void main(String args[]){
        String strStatus = "0";
        boolean status = Boolean.parseBoolean(strStatus);
        System.out.println(status);
    }
}

```

```

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques1.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques1
false

```

→ 0 and 1 string can be converted into boolean true and false by following method:

```

class Ques1{
    public static void main(String args[]){
        String strStatus="0";
        Boolean bool1=strStatus.equals("0");
    }
}

```

```

        //Boolean bool1=Boolean.parseBoolean(strStatu
s);
        System.out.println("String to bool: "+bool1);
    }
}

```

```

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques1.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques1
String to bool: true

```

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)).

→ Boxing

```

class Ques1{
    public static void main(String args[]){
        boolean status = true;
        Boolean status1 = Boolean.valueOf(status);
        System.out.println(status1);
    }
}

```

```

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques1.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques1
true

```

f. 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)).

```

class Ques1{
    public static void main(String args[]){

```

```

        String strStatus="true";
        Boolean status1 = Boolean.valueOf(strStatus);
        System.out.println(status1);
    }
}

```

```

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques1.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques1
true

```

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

```

class Ques1{
    public static void main(String args[]){
        boolean b = true;
        //to int
        int a = Integer.parseInt(b);
        System.out.println(a);
        //error: incompatible types: boolean cannot be converted to Str
        int myInt = Boolean.valueOf(b).compareTo(true);
        System.out.println(myInt);
        //For other datatypes, conversion cannot be made directly;
    }
}

```

```

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques1.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques1
0

```

## 2. Working with java.lang.Byte

- a. Explore the Java API documentation for java.lang.Byte 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).

```
class Ques2{
    public static void main(String args[]){
        Byte b1 = 24;
        System.out.println(b1.BYTES);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques2.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques2
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).

```
class Ques2{
    public static void main(String args[]){
        Byte b1 = 24;
        System.out.println(b1.MIN_VALUE + " " + b1.MAX_VALU
E);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques2.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques2
-128 127
```

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)).

```
class Ques2{
    public static void main(String args[]){
        Byte number = 24;
        String str = Byte.toString(number);
        System.out.println(str);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques2.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques2
24
```

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)).

```
class Ques2{
    public static void main(String args[]){
        String strNumber = "12";
        Byte b1 = Byte.parseByte(strNumber);
        System.out.println(b1);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques2.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques2
12
```

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).

```
class Ques2{
    public static void main(String args[]){
        String strNumber = "Ab12Cd3";
        Byte b1 = Byte.parseByte(strNumber);
        System.out.println(b1);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques2.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques2
Exception in thread "main" java.lang.NumberFormatException: For input string: "Ab12Cd3"
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:668)
    at java.base/java.lang.Byte.parseByte(Byte.java:193)
    at java.base/java.lang.Byte.parseByte(Byte.java:219)
    at Ques2.main(Ques2.java:23)
```

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)).

```
class Ques2{
    public static void main(String args[]){
        byte number = 24;
        Byte b1 = Byte.valueOf(number);
        System.out.println(b1);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques2.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques2
24
```

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)).

```
class Ques2{
    public static void main(String args[]){
        String strNumber = "12";
        Byte b1 = Byte.valueOf(strNumber);
        System.out.println(b1);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques2.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques2
12
```

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

```
class Ques2{
    public static void main(String args[]){
        byte number = 24;
        short num1 = number;
        System.out.println(num1);
        int num5 = number;
        System.out.println(num5);
        float num2 = number;
        System.out.println(num2);
        long num3 = number;
        System.out.println(num3);
        double num4 = number;
        System.out.println(num4);

        char c = (char)number;
        System.out.println(c);
    }
}
```



```
}  
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques2  
24  
24  
24.0  
24  
24.0  
↑  
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>
```

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

- Explore the Java API documentation for java.lang.Short and observe its modifiers and super types.
- Write a program to test how many bytes are used to represent a short value using the BYTES field. (Hint: Use Short.BYTES).

```
class Ques3{  
    public static void main(String args[]){  
        //part b  
        Short s1 = 800;  
        System.out.println(s1.BYTES);  
    }  
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques3.java  
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques3  
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).

```
class Ques3{
    public static void main(String args[]){
        Short b1 = 800;
        System.out.println(b1.MIN_VALUE + " " + b1.MAX_VAL
UE);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques3.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques3
-32768 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)).

```
class Ques3{
    public static void main(String args[]){
        Short number = 800;
        String str = Short.toString(number);
        System.out.println(str);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques3
800
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>
```

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)).

```
class Ques3{
    public static void main(String args[]){
        String strNumber = "1222";
        Short b1 = Short.parseShort(strNumber);
        System.out.println(b1);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques3.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques3
1222
```

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).

```
class Ques3{
    public static void main(String args[]){
        String strNumber = "Ab12Cd3";
        String b1 = String.parseByte(strNumber);
        System.out.println(b1);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques3.java
Ques3.java:23: error: cannot find symbol
        String b1 = String.parseByte(strNumber);
                             ^
symbol:   method parseByte(String)
location: class String
1 error
```

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)).

```
class Ques3{
    public static void main(String args[]){
        short number = 1124;
        Short b1 = Short.valueOf(number);
        System.out.println(b1);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques3.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques3
1124
```

i 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)).

```
class Ques3{
    public static void main(String args[]){
        short number = 1000;
        byte num1 = (byte)number;
        System.out.println(num1);
        int num5 = number;
        System.out.println(num5);
        float num2 = number;
        System.out.println(num2);
        long num3 = number;
        System.out.println(num3);
        double num4 = number;
        System.out.println(num4);
        char c = (char)number;
    }
}
```

```
        System.out.println(c);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques3
-24
1000
1000.0
1000
1000.0
?
```

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

a. Explore the Java API documentation for java.lang.Integer 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).

```
class Ques4{
    public static void main(String args[]){
        Integer s1 = 80000;
        System.out.println(s1.BYTES);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques4
4

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 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).

```
class Ques4{
    public static void main(String args[]){
        Integer b1 = 80000;
        System.out.println(b1.MIN_VALUE + " " + b1.MAX_VALU
E);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques4.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques4
-2147483648 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)).

```
class Ques4{
    public static void main(String args[]){
        Integer b1 = 80000;
        String str = Integer.toString(b1);
        System.out.println(str);
    }
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>javac Ques4.java
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques4
80000
```

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)).

```
class Ques4{
    public static void main(String args[]){
        String strNumber = "5678";
        int parsedNumber = Integer.parseInt(strNumber);
        System.out.println(parsedNumber);
    }
}
```

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).

```
class Ques4{
    public static void main(String args[]){
        String str = "Ab12Cd3";
        int ParsedNumber = Integer.parseInt(str);
        System.out.println(ParsedNumber);
    }
}
```

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)).

```
class Ques4{
    public static void main(String args[]){
        int num = 7890;
        Integer num1 = Integer.valueOf(num);
        System.out.println(num1);
    }
}
```

```
    }  
}
```

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)`).

```
class Ques4{  
    public static void main(String args[]){  
        String str = "1234";  
        Integer num = Integer.valueOf(strNumberValue);  
        System.out.println(num);  
    }  
}
```

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)`).

```
class Ques4{  
    public static void main(String args[]){  
        int a = 10;  
        int b = 20;  
        int sum = Integer.sum(a, b);  
        System.out.println(sum);  
    }  
}
```

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)`).



```

class Ques4{
    public static void main(String args[]){
        int a = 10;
        int b = 20;
        int min = Integer.min(a, b);
        int max = Integer.max(a, b);
        System.out.println(min);
        System.out.println(max);
    }
}

```

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)`).

```

class Ques4{
    public static void main(String args[]){
        int value = 7;
        String binary = Integer.toBinaryString(value);
        String octal = Integer.toOctalString(value);
        String hex = Integer.toHexString(value);
        System.out.println(binary);
        System.out.println(octal);
        System.out.println(hex);
    }
}

```

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

```

class Ques4{
    public static void main(String args[]){

```

```

        int num = 1000;
        byte byteValue = (byte) num;
        float floatValue = num;
        long longValue = num;
        double doubleValue = num;
        char charValue = (char) num;

        System.out.println(byteValue);
        System.out.println(floatValue);
        System.out.println(longValue);
        System.out.println(doubleValue);
        System.out.println(charValue);
    }
}

```

## 5. Working with java.lang.Long

```

class Ques5 {
    public static void main(String[] args) {
        // part b
        // Print the number of bytes used to represent a long value
        Long num = 200000001;
        System.out.println(num.BYTES);

        // part c
        // Print the minimum and maximum values of long
        System.out.println(Long.MIN_VALUE);
        System.out.println(Long.MAX_VALUE);

        // part d
        // Declare a long variable and convert it to a String
    }
}

```

```

ng using toString method
    long number = 123456789L;
    String str = Long.toString(number);
    System.out.println(str);

    // part e
    // Convert a String to a long value using parseLong
method
    String strNumber = "9876543210";
    long number = Long.parseLong(strNumber);
    System.out.println(number);

    // part f
    // Attempt to convert a String with non-numeric cha
racters to a long value
    String str1 = "Ab12Cd3";
    long number1 = Long.parseLong(strInvalidNumber);
    System.out.println(number1);

    // part g
    // Convert a long value to its corresponding wrappe
r class using valueOf method
    long num = 1234567890L;
    Long wrapper = Long.valueOf(num);
    System.out.println("Part g: Long to Long wrapper: "
+ wrapper);

    // part h
    // Convert a String with a long value to its corres
ponding wrapper class using valueOf method
    String str2 = "1234567890";
    Long number2 = Long.valueOf(str2);
    System.out.println("Part h: String to Long wrapper:
" + number2);

    // part i

```

```

        // Declare two long variables and add them using the
        Long.sum method
        long a = 1123L;
        long b = 9845L;
        long sum = Long.sum(a, b);
        System.out.println(sum);

        // part j
        // Find the minimum and maximum values between two
        long variables
        long min = Long.min(a, b);
        long max = Long.max(a, b);
        System.out.println(min);
        System.out.println(max);

        // part k
        // Convert a long value to binary, octal, and hexadecimal
        strings
        long value = 7L;
        String binaryString = Long.toBinaryString(value);
        String octalString = Long.toOctalString(value);
        String hexString = Long.toHexString(value);
        System.out.println(binaryString);
        System.out.println(octalString);
        System.out.println(hexString);

        // part l
        // Experiment with converting a long value into other
        primitive types or vice versa
        long num = 100000L;
        int intValue = (int) num;
        float floatValue = num;
        double doubleValue = num;
        short shortValue = (short) num;
        byte byteValue = (byte) num;
        char charValue = (char) num;

```

```

        System.out.println(intValue);
        System.out.println(floatValue);
        System.out.println(doubleValue);
        System.out.println(shortValue);
        System.out.println(byteValue);
        System.out.println(charValue);
    }
}

```

## 6. Working with java.lang.Float

```

class Ques6 {
    public static void main(String[] args) {
        // part b
        // Print the number of bytes used to represent a float value
        System.out.println(Float.BYTES);

        // part c
        // Print the minimum and maximum values of float
        System.out.println(Float.MIN_VALUE);
        System.out.println(Float.MAX_VALUE);

        // part d
        // Declare a float variable and convert it to a String using toString method
        float num = 123.45f;
        String str = Float.toString(num);
        System.out.println(str);

        // part e
    }
}

```

```

        // Convert a String to a float value using parseFloat method
        String strNum = "678.90";
        float parsedNum = Float.parseFloat(strNum);
        System.out.println(parsedNum);

        // part f
        // Attempt to convert a String with non-numeric characters to a float value
        String str = "Ab12Cd3";
        float Num = Float.parseFloat(str);
        System.out.println("Part f: String to Float: " + Num);

        // part g
        // Convert a float value to its corresponding wrapper class using valueOf method
        float numVal = 345.67f;
        Float w = Float.valueOf(numVal);
        System.out.println(w);

        // part h
        // Convert a String with a float value to its corresponding wrapper class using valueOf method
        String strNumVal = "456.78";
        Float ws = Float.valueOf(strNumVal);
        System.out.println(ws);

        // part i
        // Declare two float variables and add them using the Float.sum method
        float a = 112.3f;
        float b = 984.5f;
        float sum = Float.sum(a, b);
        System.out.println(sum);

```

```

        // part j
        // Find the minimum and maximum values between two
float variables
        float min = Float.min(112.2f, 556.6f);
        float max = Float.max(112.2f, 556.6f);
        System.out.println(min);
        System.out.println(max);

        // part k
        // Find the square root of a float value
        float numSqrt = -25.0f;
        double sqrt = Math.sqrt(numSqrt); // Math.sqrt returns a double
        System.out.println(sqrt);

        // part l
        // Divide two float variables with the same value,
0.0f
        float zero1 = 0.0f;
        float zero2 = 0.0f;
        float divisionResult = zero1 / zero2;
        System.out.println(divisionResult);

        // part m
        // Experiment with converting a float value into other primitive types or vice versa
        float floatNum = 1234.56f;
        int intValue = (int) floatNum;
        long longValue = (long) floatNum;
        double doubleValue = floatNum;
        short shortValue = (short) floatNum;
        byte byteValue = (byte) floatNum;

        System.out.println(intValue);
        System.out.println(longValue);
        System.out.println(doubleValue);

```

```
        System.out.println(shortValue);
        System.out.println(byteValue);
    }
}
```

---

## 7. Working with java.lang.Double

```
class Ques7 {
    public static void main(String[] args) {
        // part b:
        //Write a program to test how many bytes are used
        to represent a double value using the BYTES field.
        System.out.println(Double.BYTES);

        // part c:
        //Write a program to find the minimum and maximum
        values of double using the MIN_VALUE and MAX_VALUE fields.
        System.out.println(Double.MIN_VALUE);
        System.out.println(Double.MAX_VALUE);

        // part d:
        //Declare a method-local variable number of type d
        ouble with some value and convert it to a String using the
        toString method.
        double num = 123.456;
        String str = Double.toString(num);
        System.out.println(str);

        // part e:
        //Declare a method-local variable strNumber of typ
        e String with some value and convert it to a double value
        using the parseDouble method.
```



```

String strNum = "789.101";
double parsedNum = Double.parseDouble(strNum);
System.out.println(parsedNum);

// part f:
//Declare a method-local variable strNumber of type
//String with the value "Ab12Cd3" and attempt to convert it
//to a double value.
String strInvalid = "Ab12Cd3";
double invalidParsedNum = Double.parseDouble(strInvalid);
System.out.println(invalidParsedNum);

// part g:
//Declare a method-local variable number of type double
//with some value and convert it to the corresponding
//wrapper class using Double.valueOf().
double numVal = 345.678;
Double wrapper = Double.valueOf(numVal);
System.out.println(wrapper);

// part 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().
String strNumVal = "456.789";
Double wrapperFromStr = Double.valueOf(strNumVal);
System.out.println(wrapperFromStr);

// part i:
//Declare two double variables with values 112.3 and
//984.5, and add them using a method from the Double
//class.
double a = 112.3;
double b = 984.5;
double sum = Double.sum(a, b);

```

```

        System.out.println(sum);

        // part j:
        //Declare two double variables with values 112.2 and 556.6, and find the minimum and maximum values using the Double class.
        double min = Double.min(112.2, 556.6);
        double max = Double.max(112.2, 556.6);
        System.out.println(min);
        System.out.println(max);

        // part k:
        //Declare a double variable with the value -25.0.
        Find the square root of this value.
        double numSqrt = -25.0;
        double sqrt = Math.sqrt(numSqrt);
        System.out.println(sqrt);

        // part l:
        //Declare two double variables with the same value, 0.0, and divide them.
        double zero1 = 0.0;
        double zero2 = 0.0;
        double divisionResult = zero1 / zero2;
        System.out.println(divisionResult);

        // part m:
        //Experiment with converting a double value into other primitive types or vice versa.
        double num = 1234.56;
        int intValue = (int) num;
        long longValue = (long) num;
        float floatValue = (float) num;
        short shortValue = (short) num;
        byte byteValue = (byte) num;

```

```

        System.out.println(intValue);
        System.out.println(longValue);
        System.out.println(floatValue);
        System.out.println(shortValue);
        System.out.println(byteValue);
    }
}

```

## 8. Conversion between Primitive Types and Strings

Initialize a variable of each primitive type with a user-defined value and convert it into String:

- o First, use the toString method of the corresponding wrapper class. (e.g., Integer.toString()).
- o Then, use the valueOf method of the String class. (e.g., String.valueOf()).

```

class Ques8{
    public static void main(String args[]){
        Byte b = 20;
        Short s = 200;
        Integer i = 2000;
        Long l = 200001;
        Float f = 20.0f;
        Double d = 20.5;

        System.out.println(Byte.toString(b));
        System.out.println(Short.toString(s));
        System.out.println(Integer.toString(i));
        System.out.println(Long.toString(l));
        System.out.println(Float.toString(f));
        System.out.println(Double.toString(d));

        System.out.println(String.valueOf(b));
    }
}

```

```

        System.out.println(String.valueOf(s));
        System.out.println(String.valueOf(i));
        System.out.println(String.valueOf(l));
        System.out.println(String.valueOf(f));
        System.out.println(String.valueOf(d));
    }
}

```

```

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques8
20
200
2000
20000
20.0
20.5
20
200
2000
20000
20.0
20.5

```

## 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).

```

class Dtypes{
    byte b;
    short s;
    int i;
    long l;
    float f;
    double d;
}

```

```

        String str;
    }

    class Ques9{
        public static void main(String args[]){
            Dtypes dt = new Dtypes();

            System.out.println(dt.b);
            System.out.println(dt.s);
            System.out.println(dt.i);
            System.out.println(dt.l);
            System.out.println(dt.f);
            System.out.println(dt.d);
            System.out.println(dt.str);
        }
    }
}

```

```

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques9
0
0
0
0
0.0
0.0
null

```

#### 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).

```

class Ques10{
    public static void main(String args[]){
        int num1 = Integer.parseInt(args[0]);

```

```
String op = args[1];
int num2 = Integer.parseInt(args[2]);

switch (op){
    case "+":
        System.out.println(num1 + num2);
        break;
    case "-":
        System.out.println(num1 - num2);
        break;
    case "*":
        System.out.println(num1 * num2);
        break;
    case "/":
        System.out.println(num1 / num2);
        break;
}
}
```

```
C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques10 10 + 20
30

C:\Users\reddy\Desktop\CDAC Labs\OOPJ Module\Assignment 4>java Ques10 10 - 20
-10
```