Name	Tarun Kumar
Branch	ECE
Enrollment Number	24/11/EE/011

Q1. Develop a Java program to explore the characteristics of the four main integer primitive types: byte, short, int, and long. Begin by declaring variables of each type and assigning them their respective maximum values. Once done, print the output to observe how Java handles their representation. Extend the program to test scenarios where these variables are assigned values beyond their allowed limits, and examine any compilation or runtime responses encountered.

```
public class Problem1 {
    public static void main(String[] args){
        byte b = Byte.MAX_VALUE;        short s = Short.MAX_VALUE;
        int i = Integer.MAX_VALUE;      long l = Long.MAX_VALUE;
        System.out.println("Max value in byte datatype : "+b);
        System.out.println("Max value in short datatype : "+s);
        System.out.println("Max value in int datatype : "+i);
        System.out.println("Max value in long datatype : "+l);
            // b = 128; s = 32768; i =  2147483648; l = 9223372036854775808;  // It will throughout error as we assign
            value greater than max value of byte, short, int, long.
    }
}
```

Case I

```
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment> javac Problem1.java;java Problem1;
Max value in byte datatype : 127
Max value in short datatype : 32767
Max value in int datatype : 2147483647
Max value in long datatype : 9223372036854775807
```

Case II *=> on uncomment b = 128; s = 32768; i = 2147483648; l = 9223372036854775808*

```
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment> javac Problem1.java
Problem1.java:15: error: integer number too large
            b = 128; s = 32768; i =  2147483648; l = 9223372036854775808;
                                     ^
Problem1.java:15: error: integer number too large
            b = 128; s = 32768; i =  2147483648; l = 9223372036854775808;
                                                     ^
2 errors
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment>
```

Q2. Write a Java program to investigate the differences in precision and behavior between the float and double data types. Assign both types the same numeric values and compare their printed outputs to assess how Java handles decimal precision. Experiment with additional fractional inputs to observe rounding patterns and understand how these data types manage precision loss in mathematical contexts.

```
public class Problem2 {
    public static void main(String[] args){
        float f = 0.3333333333333333f;
        double d = 0.3333333333333333d;
        System.out.println("Float value : "+f+"\nDouble value : "+d+"\nfor value = 0.3333333333333333");
        f = 10/3f;      d = 10/3d;
        System.out.println("Float value : "+f+"\nDouble value : "+d+"\nfor value = 10/3");
    }
}
```

```
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment> javac Problem2.java;java Problem2;
Float value : 0.33333334
Double value : 0.3333333333333333
for value = 0.3333333333333333
Float value : 3.3333333
Double value : 3.3333333333333335
for value = 10/3
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment>
```

Q3. Construct a Java program utilizing the char primitive type to represent characters and symbols through Unicode escape sequences. Declare multiple character variables and display their output to understand character representation. Incorporate Unicode values to print symbols and analyze how character arithmetic, such as incrementing characters or converting them to integer values, behaves within the Java environment.

```
public class Problem3 {
    public static void main(String[] args){
        char ch1 = '\u0041',ch2 = '\u03A9'; // Unicode for 'A' && Greek capital letter Omega (Ω)
        System.out.println("Character ch1 (\\u0041): " + ch1+"\nCharacter ch4 (\\u03A9): "+ ch2);
        char letter = 'A';   System.out.println("Original letter: " + letter);
        letter++;            System.out.println("After incrementing: " + letter);
        System.out.println("ANCII value of 'A' is : "+(int)'A');
    }
}
```

```
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment> javac Problem3.java;java Problem3;
Character ch1 (\u0041): A
Character ch4 (\u03A9): Ω
Original letter: A
After incrementing: B
ANCII value of 'A' is : 65
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment>
```

Q4. Create a Java program that applies boolean logic for decision-making processes. Define several boolean variables and use logical operators like &&, ||, and ! to build compound expressions. Implement conditional statements using if and else blocks to demonstrate how boolean evaluations influence program flow and outcome based on the truth values of given conditions.

```
public class Problem4 {
   public static void main(String[] args) {
      boolean true1 = true, true2 = true, false1 = false, false2 = false;
      System.out.println("'true && true' => "+ (true && true) +"\n'true && false' => "+(true && false)+"\n'false && false' =>
         "+(false && false));
      System.out.println("'true || true' => "+ (true || true) +"\n'true || false' => "+(true || false)+"\n'false || false' => "+(false
         || false));
      System.out.println("'!true' => "+ (!true) +"\n'!false' => "+(!false));
      if(true1 && true2) System.out.println("inside if loop because 'true && true' gives true");
      else if(true1 && false1 || false2 && true2 || false1 && false2) System.out.println("'true && false' 'false && true' 'false
         && false' gives false");
   }
}
```

```
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment> javac Problem4.java;java Problem4;
'true && true' => true
'true && false' => false
'false && false' => false
'true || true' => true
'true || false' => true
'false || false' => false
'!true' => false
'!false' => true
inside if loop because 'true && true' gives true
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment>
```

Q5. Design a Java program that includes a utility class containing methods to validate whether given numbers fall within the acceptable ranges of the primitive types byte, short, int, and long. Test each method with various input values to ensure accurate validation. Finally, enhance the program to accept user input, thereby enabling dynamic type checks and demonstrating practical applications of range enforcement.

```
public class Problem5 {
   public static void main(String[] args){
      Scanner sc = new Scanner(System.in);
      ValidateRange obj = new ValidateRange();
      long[] arr = { 127L, -128L, 34334322L,343483648377L};
      for(long n : arr){
         obj.fun(n);
      }
      System.out.print("Enter any custom value for check it : ");
      try {
```

```
            long temp = sc.nextLong();
            obj.fun(temp);
        } catch (Exception e) {
            System.out.println("Enter valid number");
        }
        sc.close();
    }
}
```

```
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment> javac Problem5.java;java Problem5;
127 : fits Byte range
127 : fits Short range
127 : fits int range
127 : fits Long range
-128 : fits Byte range
-128 : fits Short range
-128 : fits int range
-128 : fits Long range
34334322 : fits int range
34334322 : fits Long range
343483648377 : fits Long range
Enter any custom value for check it : -183747638362
-183747638362 : fits Long range
PS C:\Users\TARUN\Desktop\Sem_3\JAVA\Assignment>
```