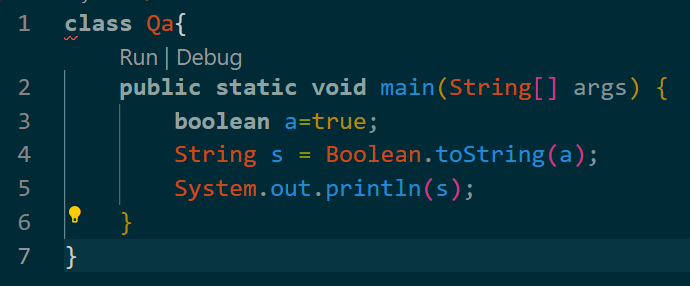
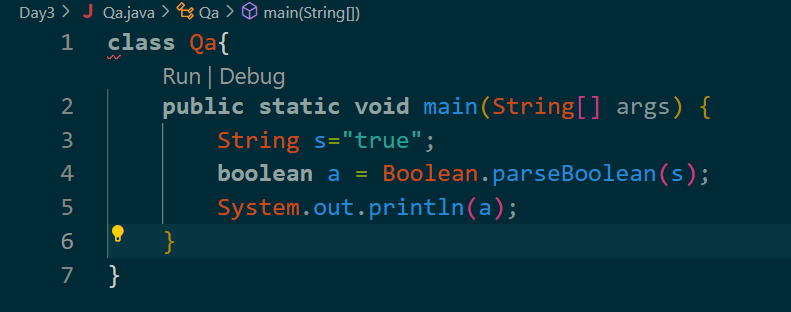
***Assignment -2***

Date – 5/9/2024

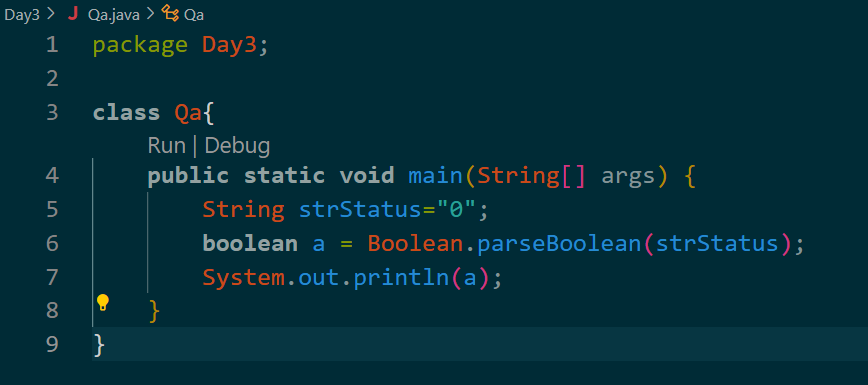
1. **Working with java.lang.Boolean**
2. Declare a method-local variable status of type boolean with the value true and convert it to a String using the toString method. (Hint: Use Boolean.toString(Boolean) ).



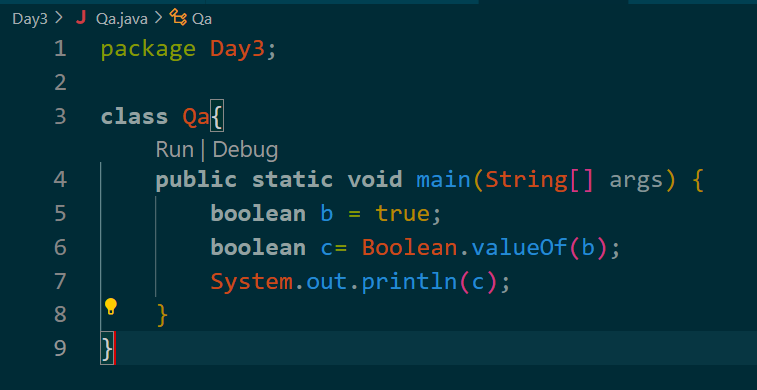
1. Declare a method-local variable strStatus of type String with the value "true" and convert it to a boolean using the parseBoolean method. (Hint: Use Boolean.parseBoolean(String)).



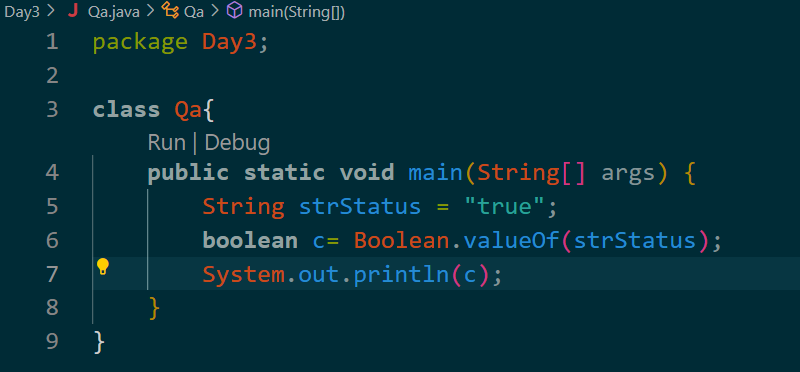
1. Declare a method-local variable strStatus of type String with the value "1" or "0" and attempt to convert it to a boolean. (Hint: parseBoolean method will not work as expected with "1" or "0").



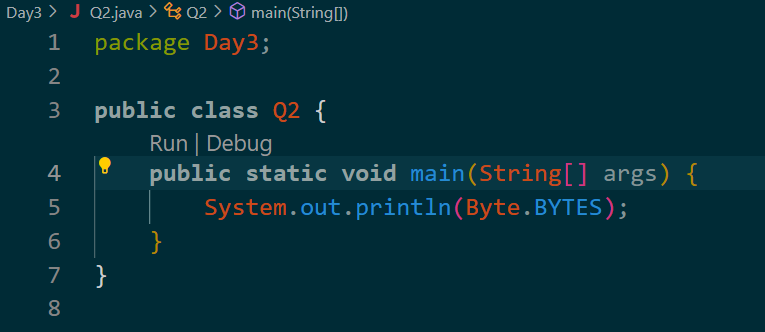
1. Declare a method-local variable status of type boolean with the value true and convert it to the corresponding wrapper class using Boolean.valueOf(). (Hint: Use Boolean.valueOf(boolean)).



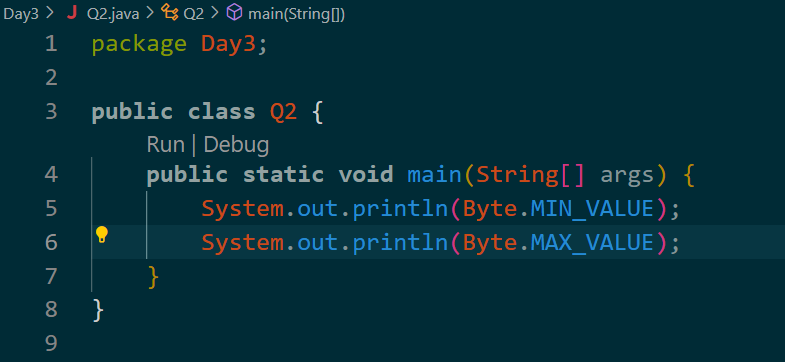
1. Declare a method-local variable strStatus of type String with the value "true" and convert it to the corresponding wrapper class using Boolean.valueOf(). (Hint: Use Boolean.valueOf(String)).



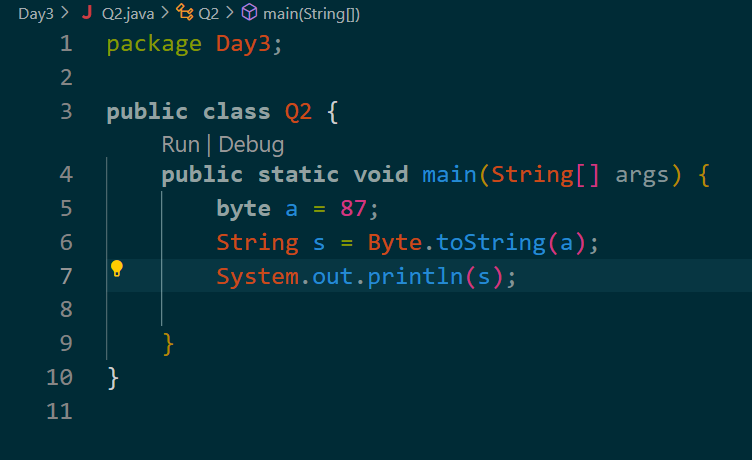
1. **Working with java.lang.Byte**
2. Write a program to test how many bytes are used to represent a byte value using the BYTES field. (Hint: Use Byte.BYTES).



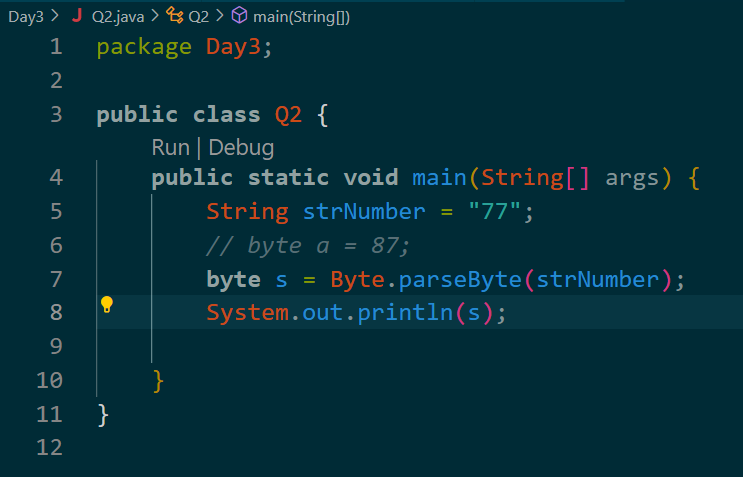
1. Write a program to find the minimum and maximum values of byte using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Byte.MIN\_VALUE and Byte.MAX\_VALUE).

****

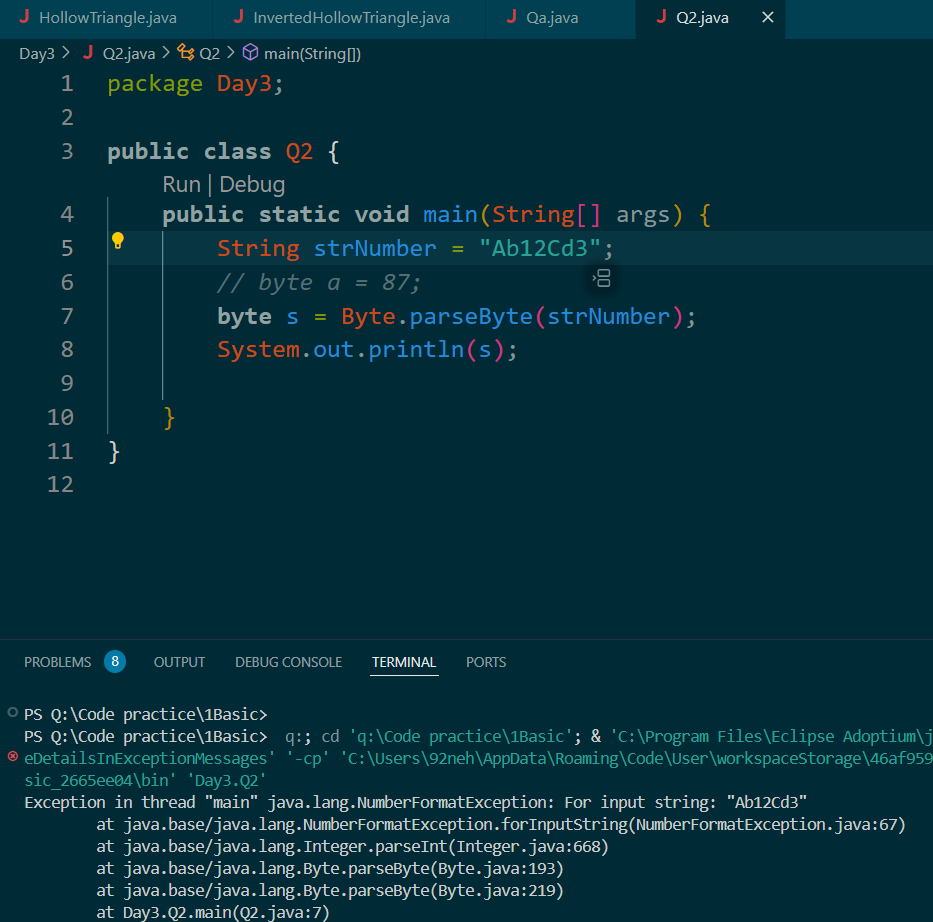
1. **.** Declare a method-local variable number of type byte with some value and convert it to a String using the toString method. (Hint: Use Byte.toString(byte)).

****

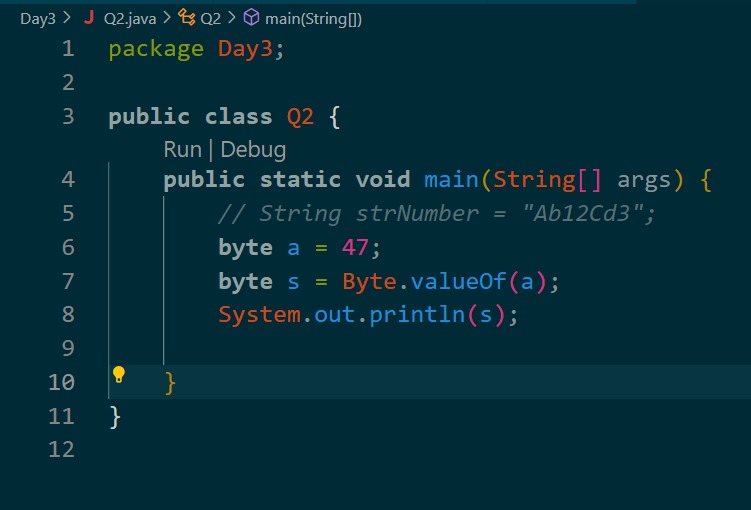
1. Declare a method-local variable strNumber of type String with some value and convert it to a byte value using the parseByte method. (Hint: Use Byte.parseByte(String)).



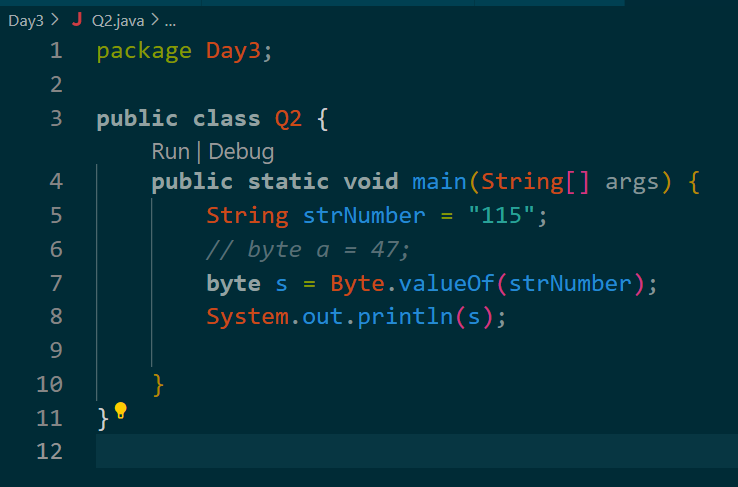
1. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a byte value. (Hint: parseByte method will throw a NumberFormatException).



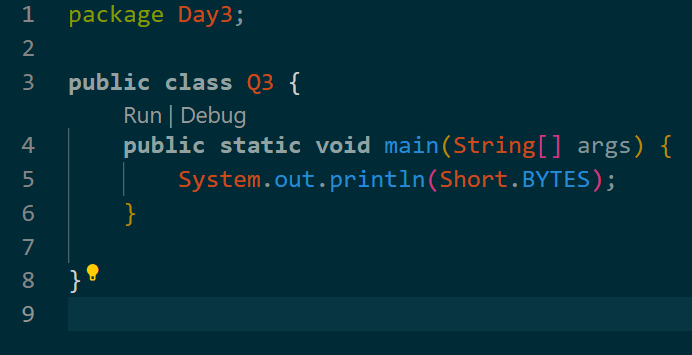
1. Declare a method-local variable number of type byte with some value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(byte)).



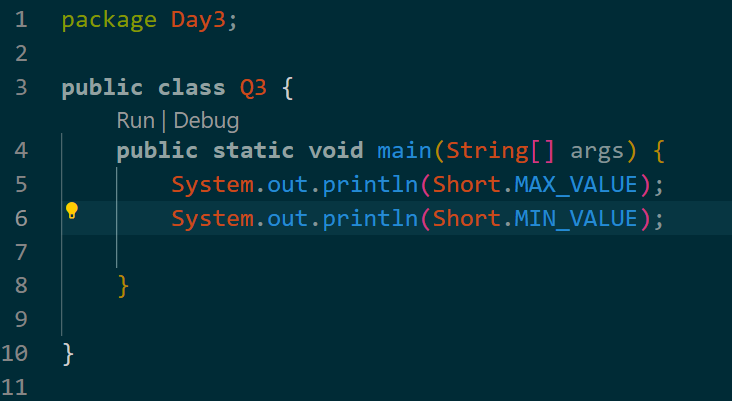
1. Declare a method-local variable strNumber of type String with some byte value and convert it to the corresponding wrapper class using Byte.valueOf(). (Hint: Use Byte.valueOf(String)).



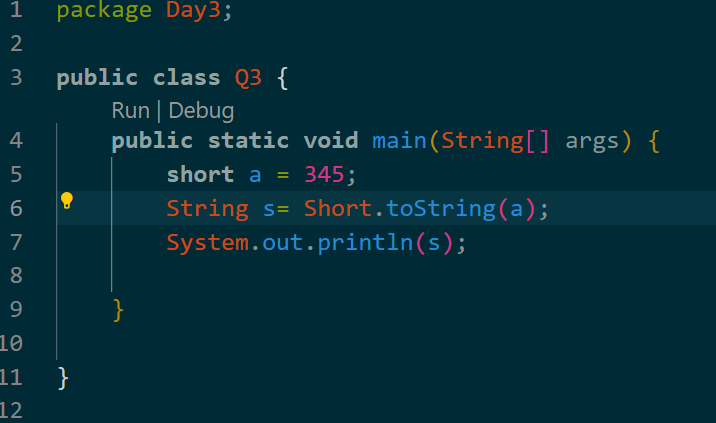
1. **Working with java.lang.Short**
2. Write a program to test how many bytes are used to represent a short value using the BYTES field. (Hint: Use Short.BYTES).



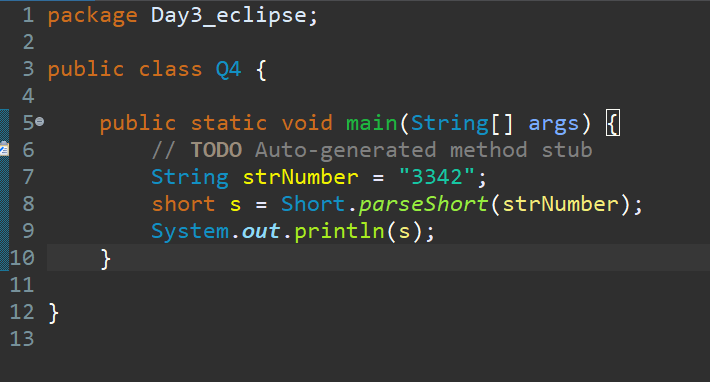
1. Write a program to find the minimum and maximum values of short using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Short.MIN\_VALUE and Short.MAX\_VALUE).



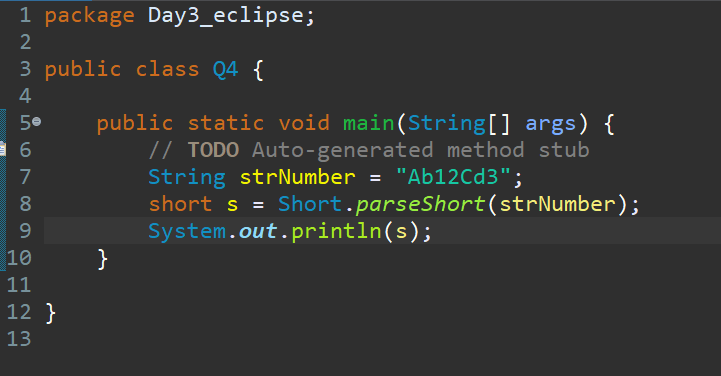
1. Declare a method-local variable number of type short with some value and convert it to a String using the toString method. (Hint: Use Short.toString(short)).



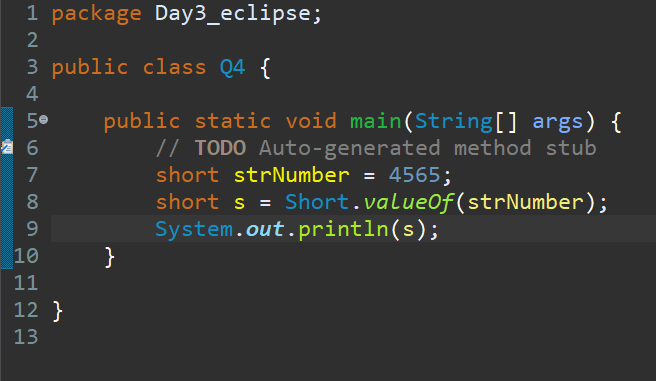
1. Declare a method-local variable strNumber of type String with some value and convert it to a short value using the parseShort method. (Hint: Use Short.parseShort(String)).



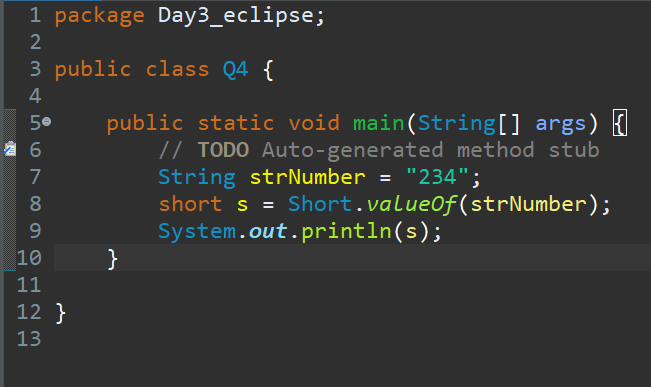
1. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a short value. (Hint: parseShort method will throw a NumberFormatException).



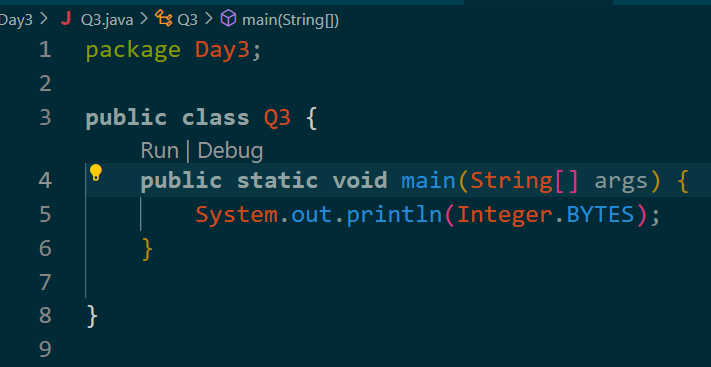
1. Declare a method-local variable number of type short with some value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(short)).



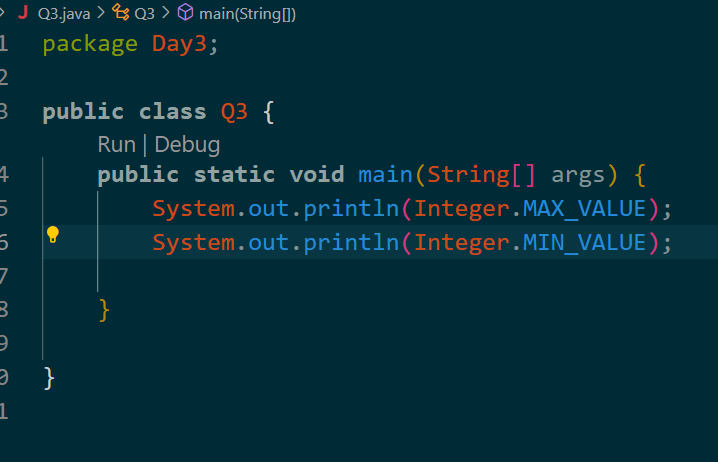
1. Declare a method-local variable strNumber of type String with some short value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Short.valueOf(String)).



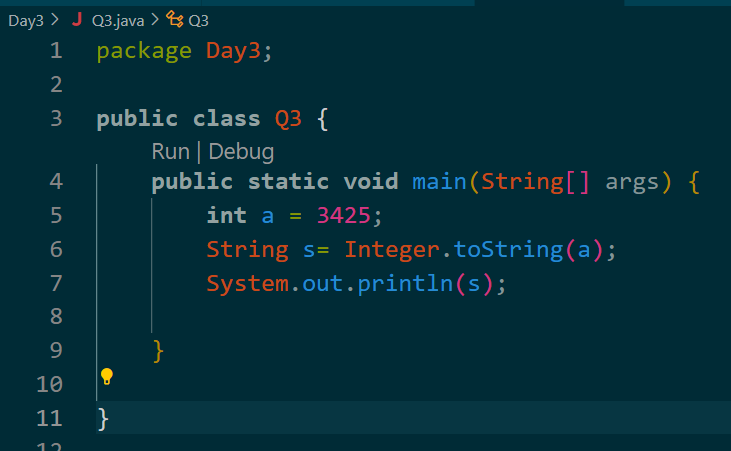
1. **Working with java.lang.Integer**
2. Write a program to test how many bytes are used to represent a Integer value using the BYTES field. (Hint: Use Integer.BYTES).



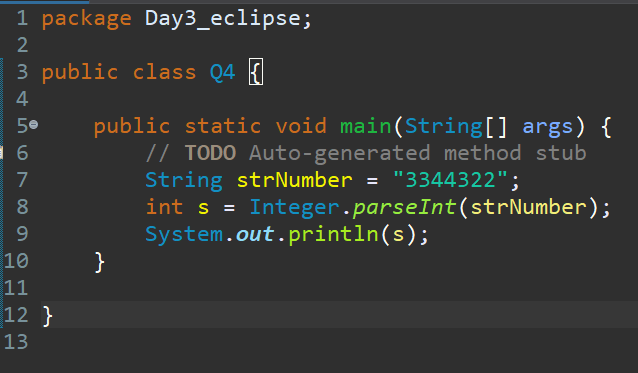
1. Write a program to find the minimum and maximum values of Integer using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Integer.MIN\_VALUE and Integer.MAX\_VALUE).

****

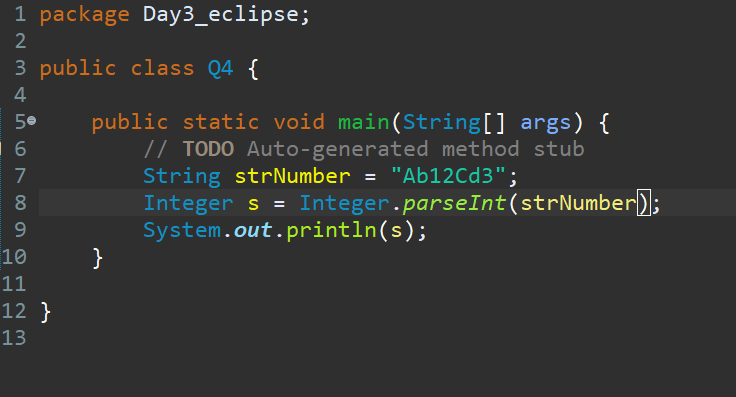
1. Declare a method-local variable number of type Integer with some value and convert it to a String using the toString method. (Hint: Use Integer.toString(short)).

****

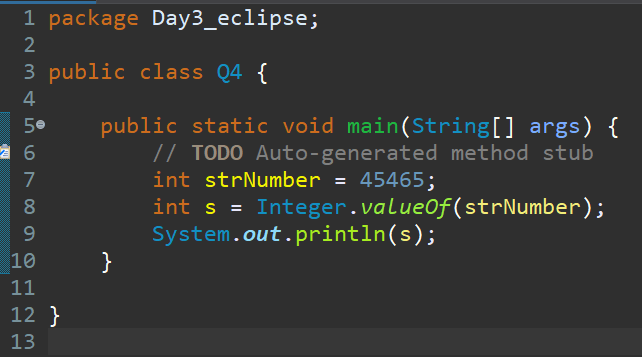
1. **Declare a method-local variable strNumber of type String with some value and convert it to a** Integer **value using the parse**Integer **method. (Hint: Use** Integer**.parseShort(String)).**

****

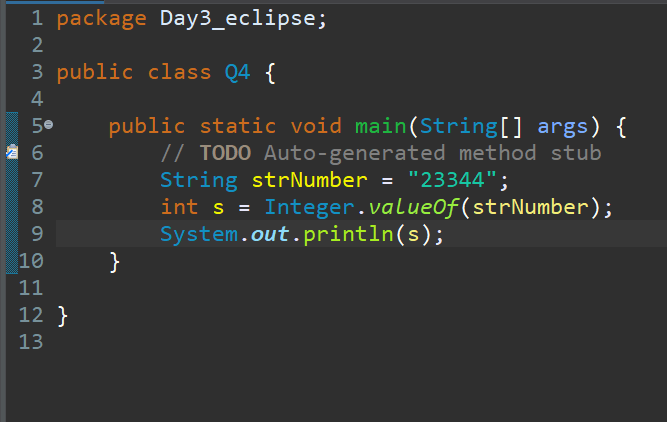
1. **Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a** Integer **value. (Hint: parse** Integer **method will throw a NumberFormatException).**

****

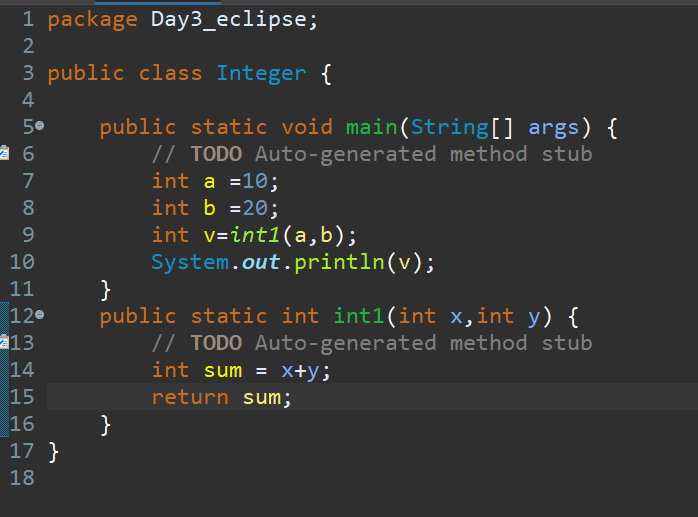
1. Declare a method-local variable number of type Integer with some value and convert it to the corresponding wrapper class using Short.valueOf(). (Hint: Use Integer.valueOf(short)).



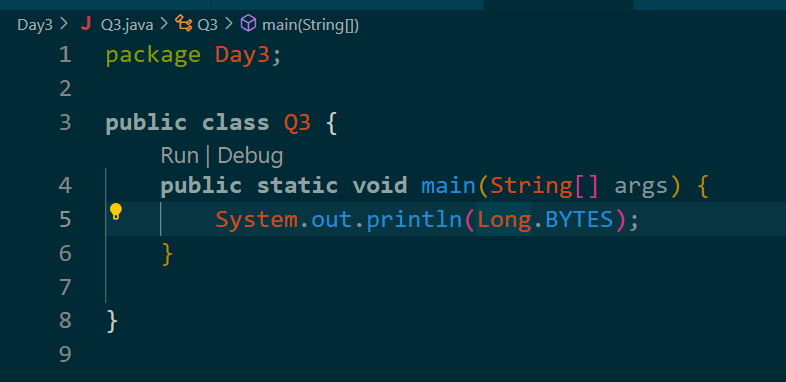
1. Declare a method-local variable strNumber of type String with some Integer value and convert it to the corresponding wrapper class using Integer.valueOf(). (Hint: Use Integer.valueOf(String)).



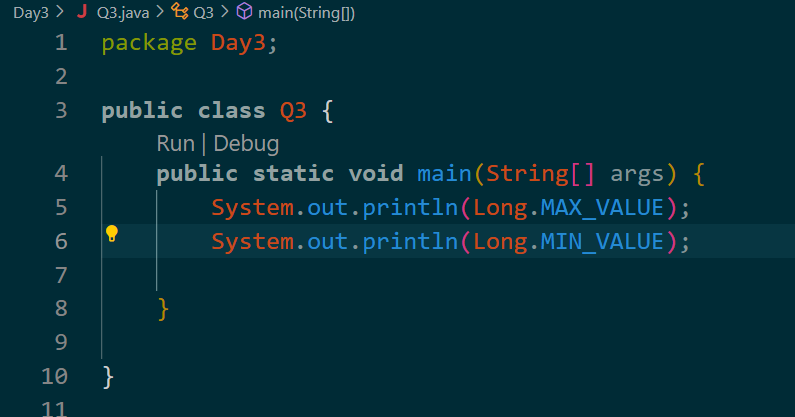
1. Declare two integer variables with values 10 and 20, and add them using a method from the Integer class. (Hint: Use Integer.sum(int, int)).



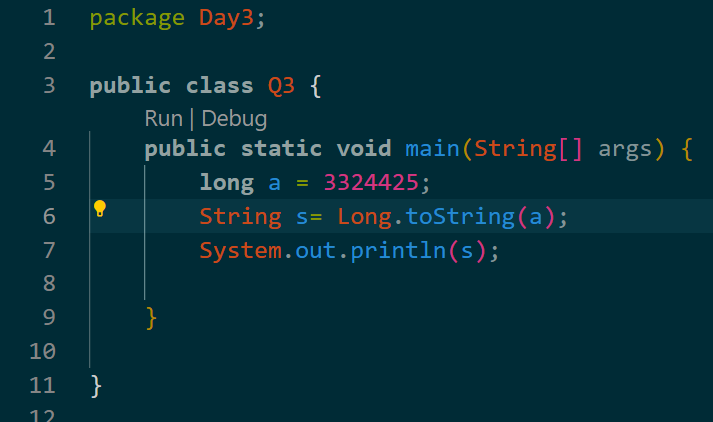
1. **Working with java.lang.Long**
2. Write a program to test how many bytes are used to represent a Long value using the BYTES field. (Hint: Use Long.BYTES).



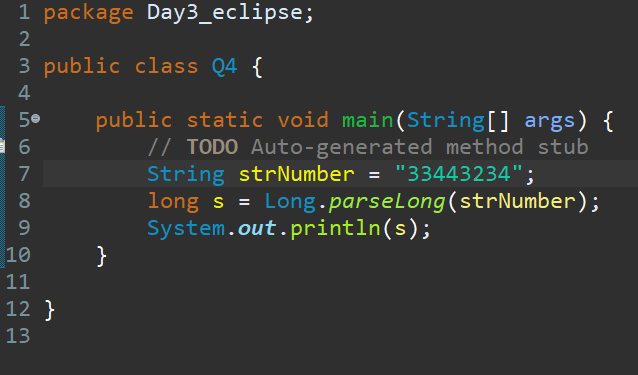
1. Write a program to find the minimum and maximum values of Long using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Long.MIN\_VALUE and Long.MAX\_VALUE).



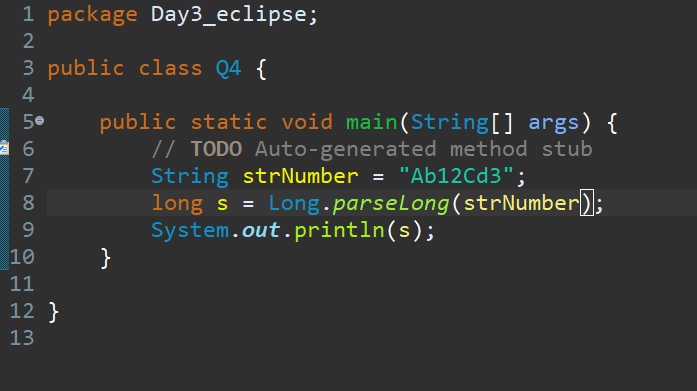
1. Declare a method-local variable number of type **Long** with some value and convert it to a String using the toString method. (Hint: Use **Long**.toString(short)).

****

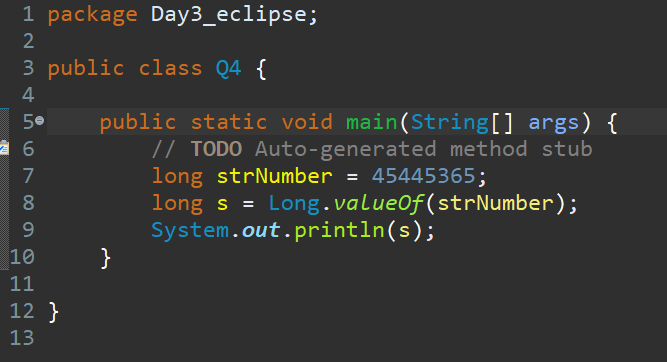
1. **Declare a method-local variable strNumber of type String with some value and convert it to a Long value using the parse Long method. (Hint: Use Short.parse Long (String)).**

****

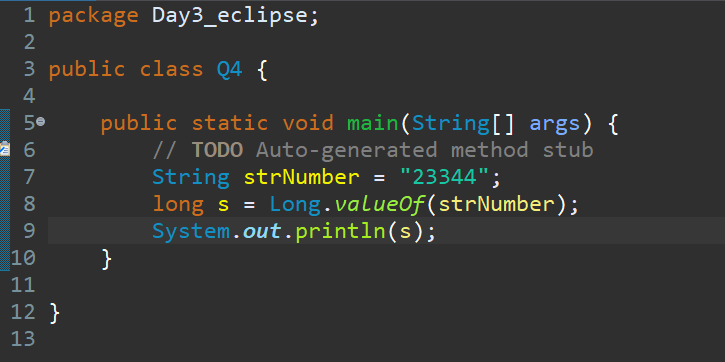
1. **Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a Long value. (Hint: parseShort method will throw a NumberFormatException).**

****

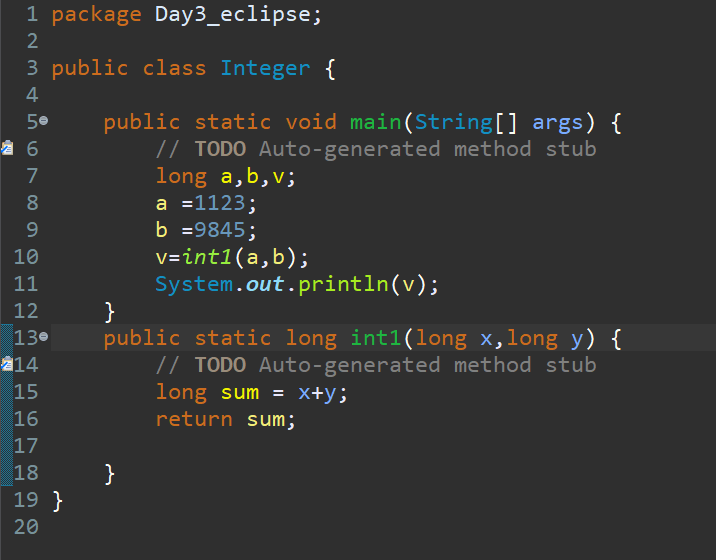
1. Declare a method-local variable number of type **Long** with some value and convert it to the corresponding wrapper class using **Long**.valueOf(). (Hint: Use **Long**.valueOf(short)).



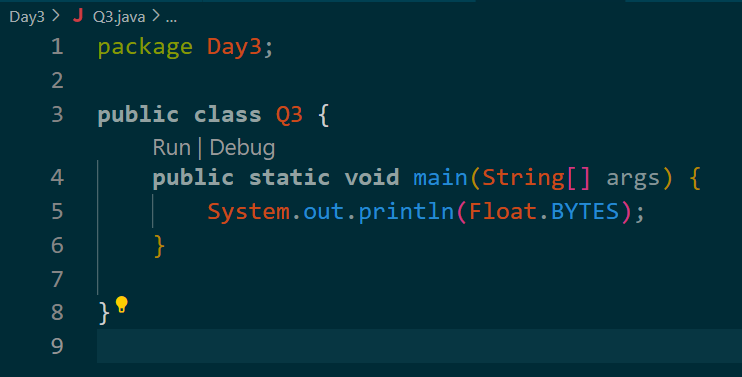
1. **.** Declare a method-local variable strNumber of type String with some **Long** value and convert it to the corresponding wrapper class using **Long**.valueOf(). (Hint: Use **Long**.valueOf(String)).



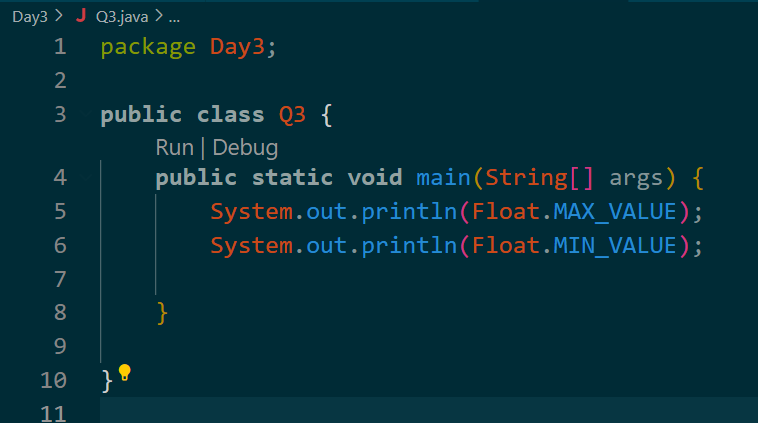
1. Declare two **Long** variables with values 1123 and 9845, and add them using a method from the **Long** class. (Hint: Use **Long**.sum(int, int)).



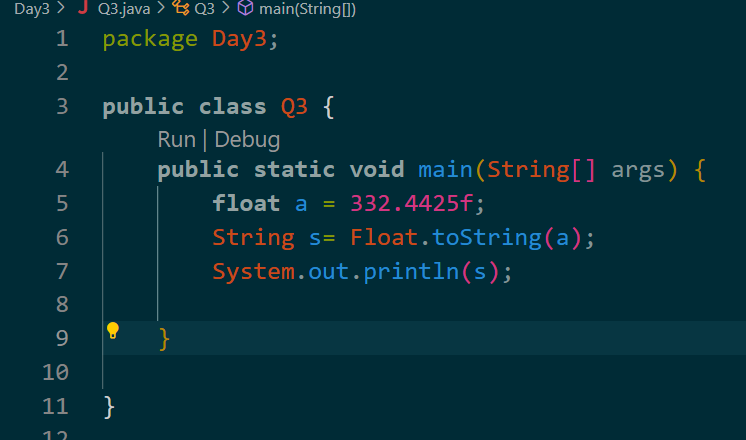
1. **Working with java.lang.Float**
2. Write a program to test how many bytes are used to represent a Float value using the BYTES field. (Hint: Use Float.BYTES).

****

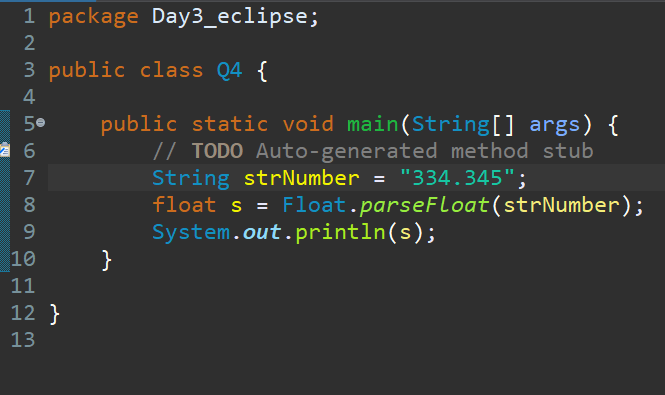
1. Write a program to find the minimum and maximum values of Float using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Float.MIN\_VALUE and Float.MAX\_VALUE).



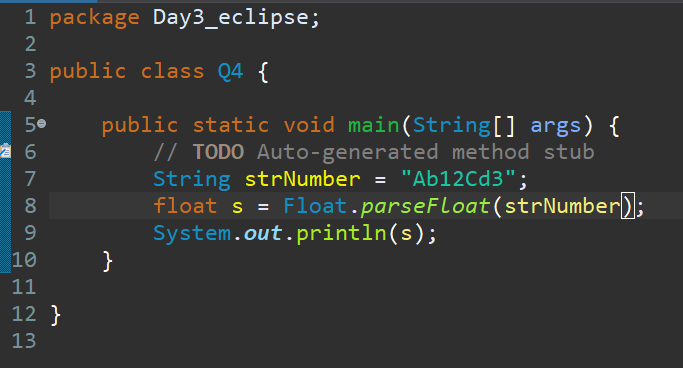
1. Declare a method-local variable number of type **Float** with some value and convert it to a String using the toString method. (Hint: Use **Float**.toString(short)).



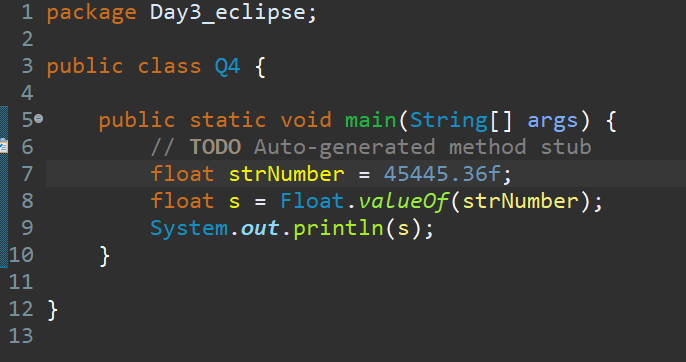
1. **Declare a method-local variable strNumber of type String with some value and convert it to a Float value using the parse Float method. (Hint: Use Float.parseShort(String)).**



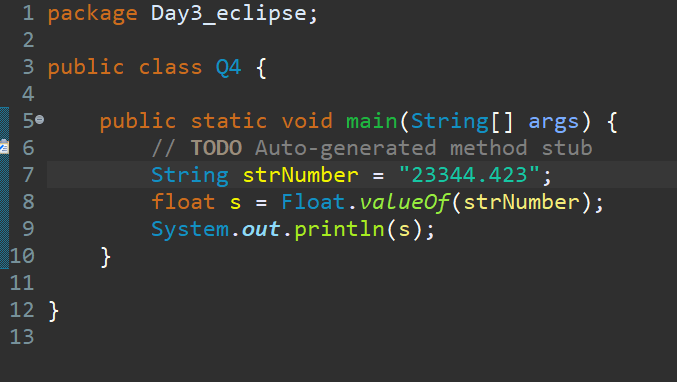
1. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a **Float** value. (Hint: parse **Float** method will throw a NumberFormatException).



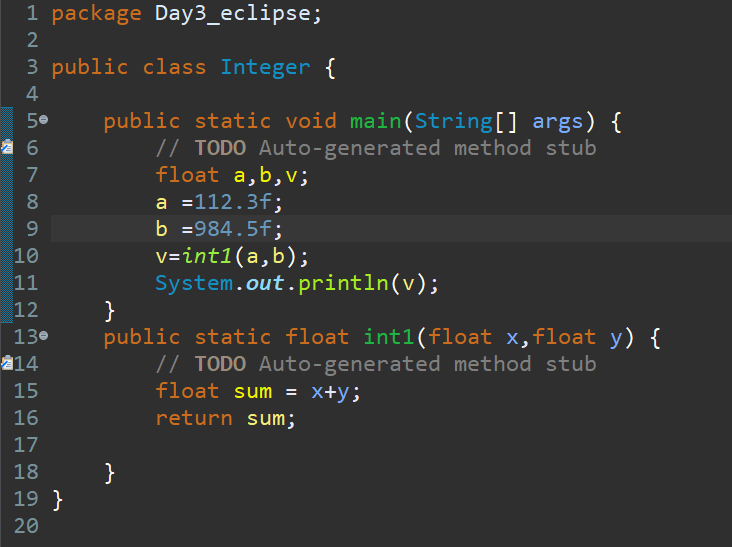
1. Declare a method-local variable number of type **Float** with some value and convert it to the corresponding wrapper class using **Float**.valueOf(). (Hint: Use **Float**.valueOf(short)).



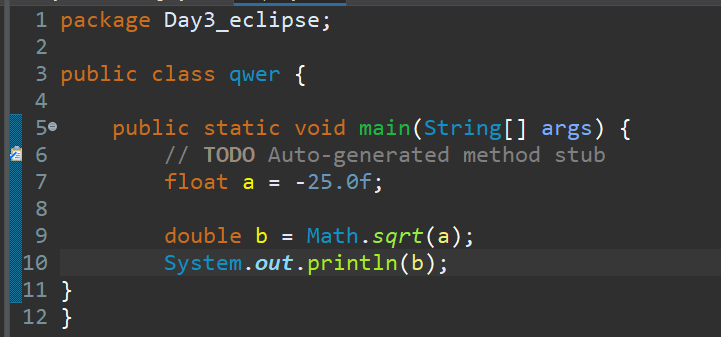
1. Declare a method-local variable strNumber of type String with some **Float** value and convert it to the corresponding wrapper class using **Float**.valueOf(). (Hint: Use **Float**.valueOf(String)).



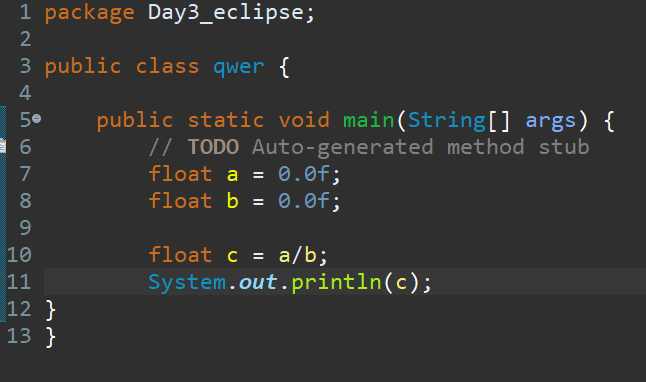
1. Declare two **Float** variables with values 112.3 and 984.5, and add them using a method from the **Float** class. (Hint: Use **Float**.sum(int, int)).



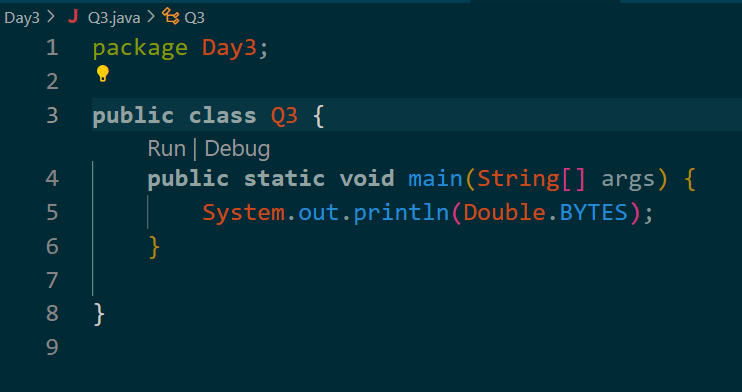
1. Declare a float variable with the value -25.0f. Find the square root of this value. (Hint: Use Math.sqrt() method).



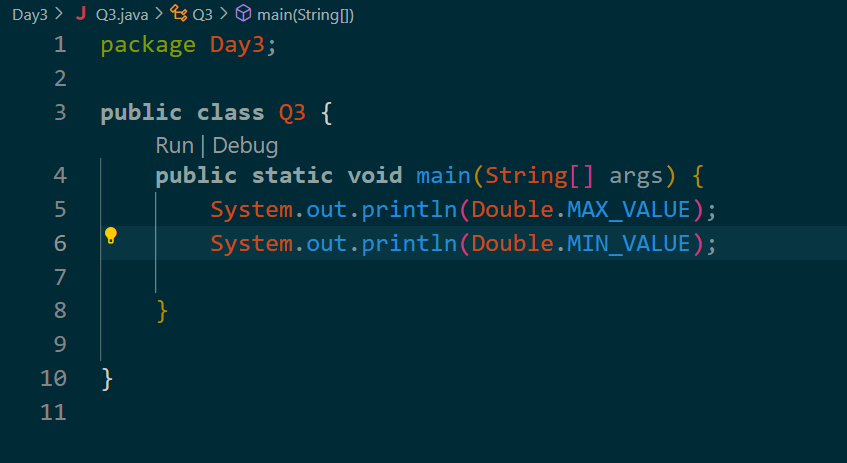
1. Declare two float variables with the same value, 0.0f, and divide them.



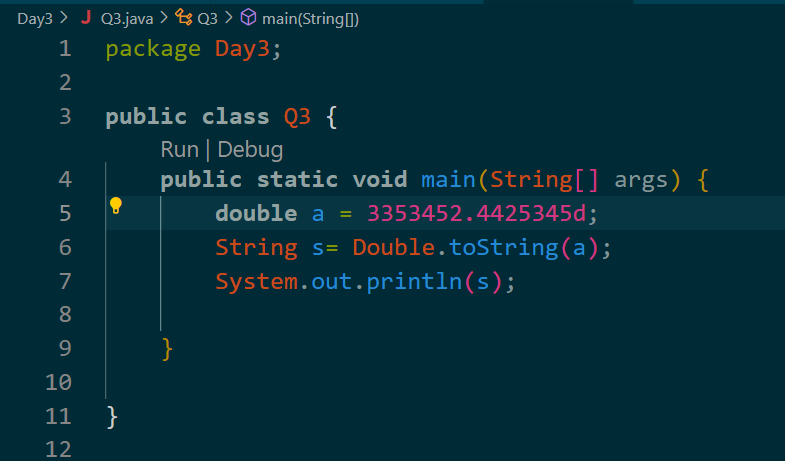
1. **Working with java.lang.Double**
2. Write a program to test how many bytes are used to represent a Double value using the BYTES field. (Hint: Use Double.BYTES).



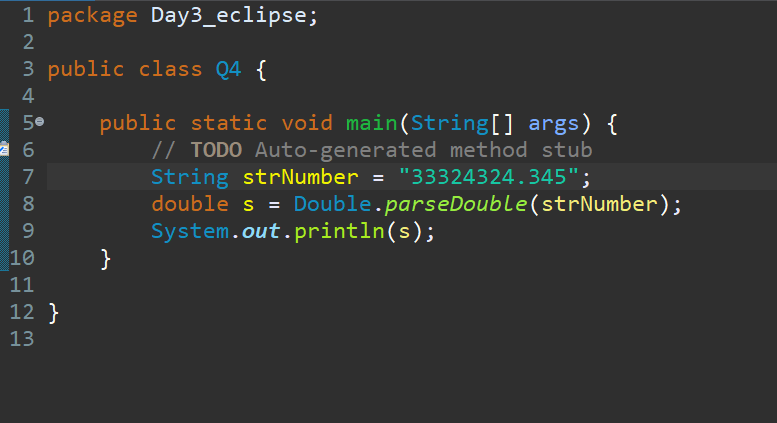
1. Write a program to find the minimum and maximum values of double using the MIN\_VALUE and MAX\_VALUE fields. (Hint: Use Double.MIN\_VALUE and Double.MAX\_VALUE).



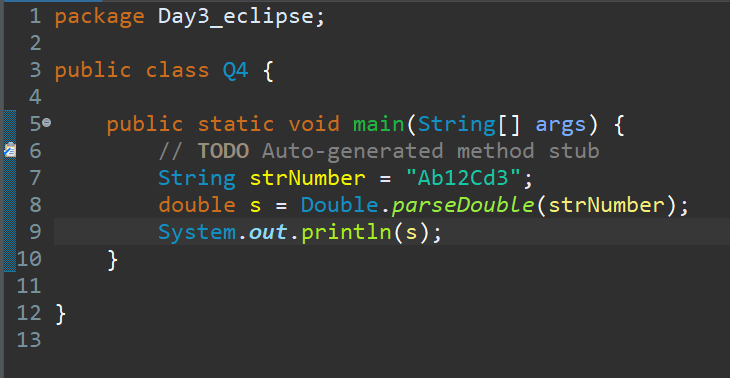
1. Declare a method-local variable number of type double with some value and convert it to a String using the toString method. (Hint: Use double.toString(short)).



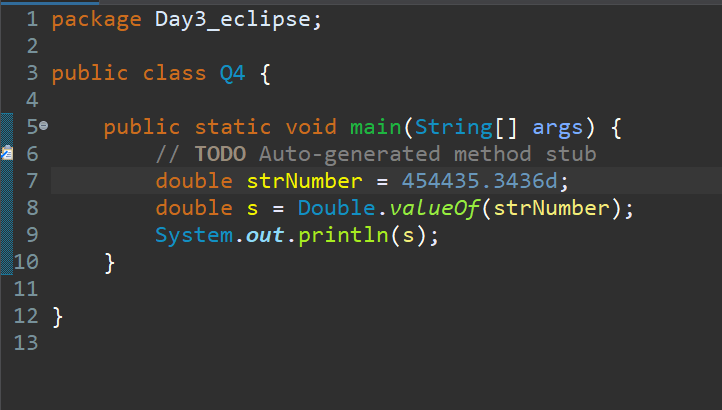
1. **Declare a method-local variable strNumber of type String with some value and convert it to a** double **value using the parse** double **method. (Hint: Use** double**.parse** double **(String)).**

****

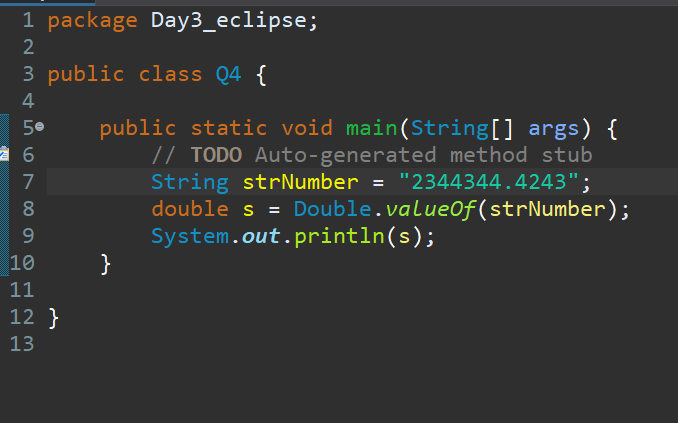
1. Declare a method-local variable strNumber of type String with the value "Ab12Cd3" and attempt to convert it to a double value. (Hint: parse double method will throw a NumberFormatException).



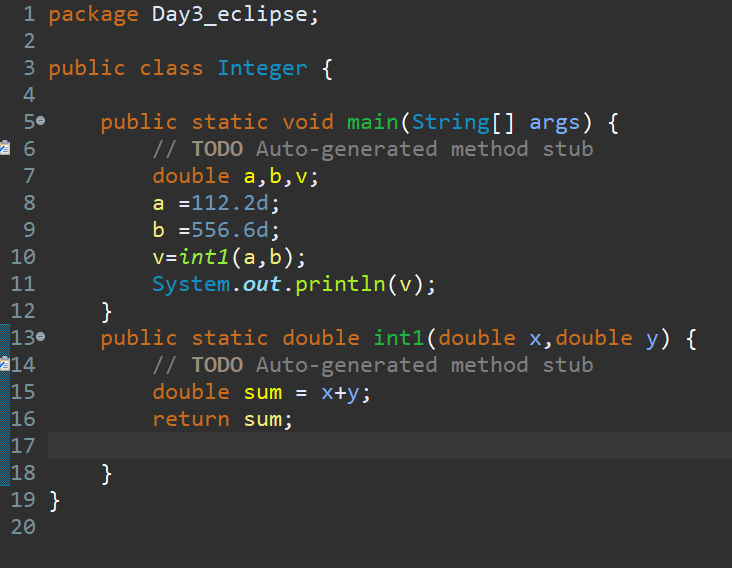
1. Declare a method-local variable number of type double with some value and convert it to the corresponding wrapper class using double.valueOf(). (Hint: Use double.valueOf(short)).



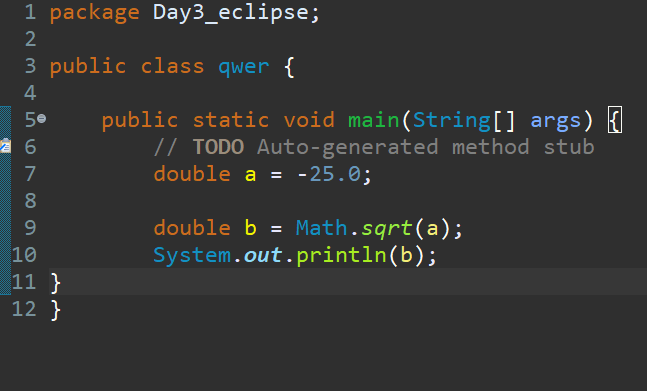
1. Declare a method-local variable strNumber of type String with some double value and convert it to the corresponding wrapper class using double.valueOf(). (Hint: Use double.valueOf(String)).



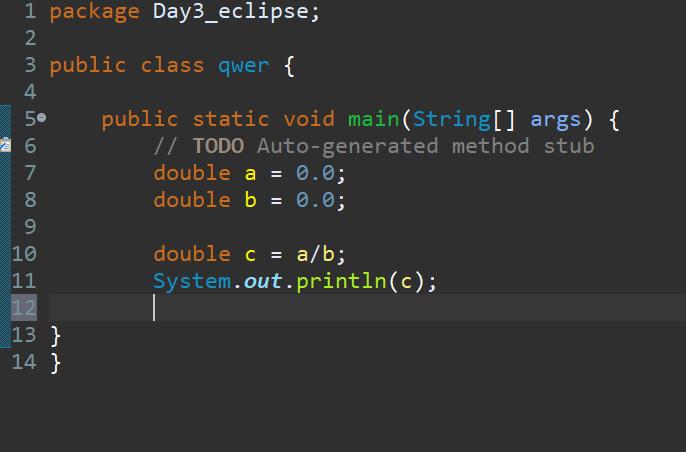
1. Declare two double variables with values 10 and 20, and add them using a method from the double class. (Hint: Use double.sum(int, int)).



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

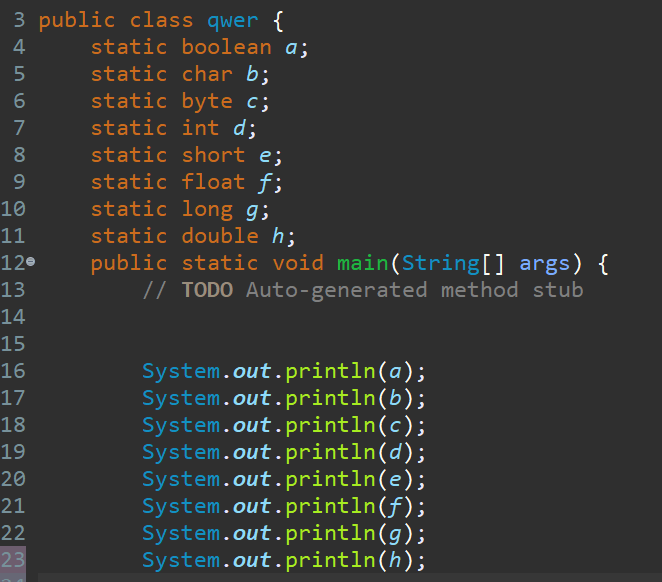


1. Declare two double variables with the same value, 0.0, and divide them. (Hint: Observe the result and any special floating-point behavior).



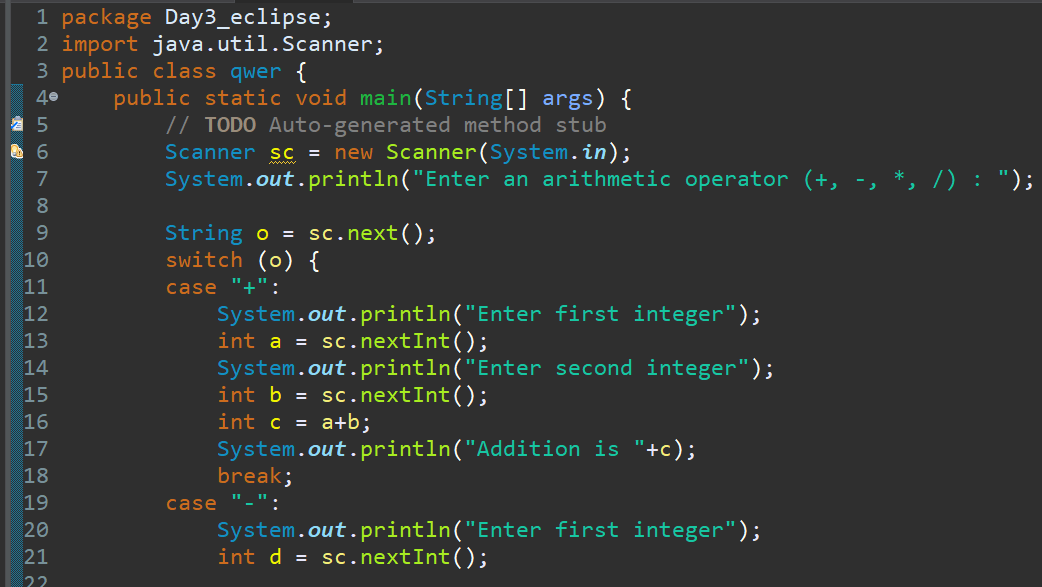
**9) Default Values of Primitive Types**

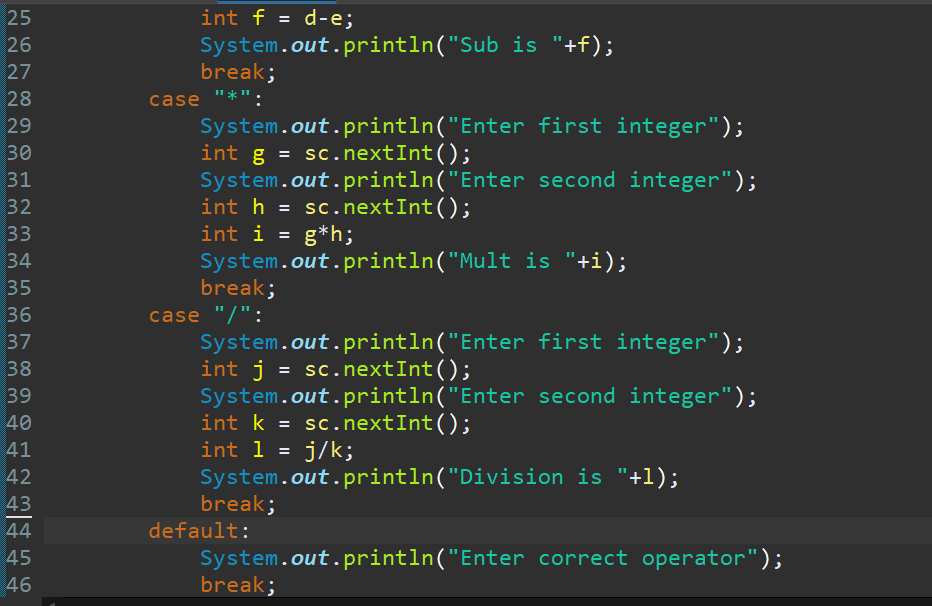
**Declare variables of each primitive type as fields of a class and check their default values. (Note: Default values depend on whether the variables are instance variables or static variables).**

****

**10. Arithmetic Operations with Command Line Input**

**Write a program that accepts two integers and an arithmetic operator (+, -, \*, /) from the command line. Perform the specified arithmetic operation based on the operator provided. (Hint: Use switch-case for operations).**

****

****