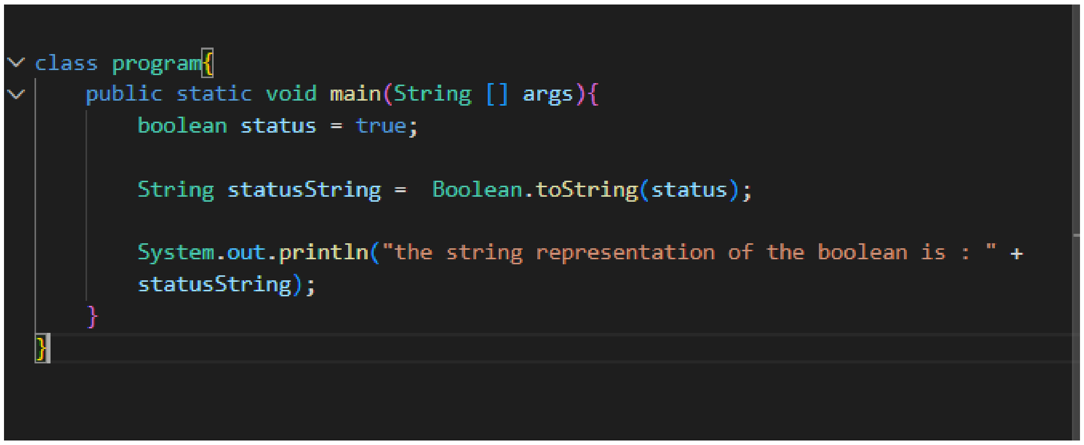
1. Working with `java.lang.Boolean`

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)`).

Ans.

```
class program{

  public static void main(String [] args){

    boolean status = true;



    String statusString =  Boolean.toString(status);



    System.out.println("the string representation of the boolean is : " + statusString);

  }

}
```

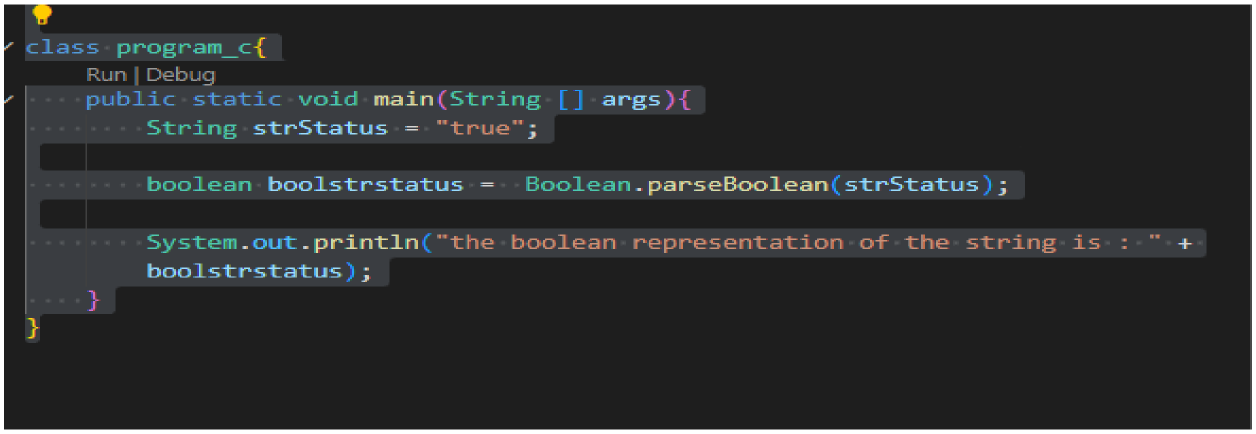


```
class program{
    public static void main(String [] args){
        boolean status = true;

        String statusString =  Boolean.toString(status);

        System.out.println("the string representation of the boolean is : " +
        statusString);
    }
}
```

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)`).

Ans.

```
class program_c{

  public static void main(String [] args){
        String strStatus = "true";


        boolean boolstrstatus =  Boolean.parseBoolean(strStatus);


        System.out.println("the boolean representation of the string is : " + boolstrstatus);

    }

}
```

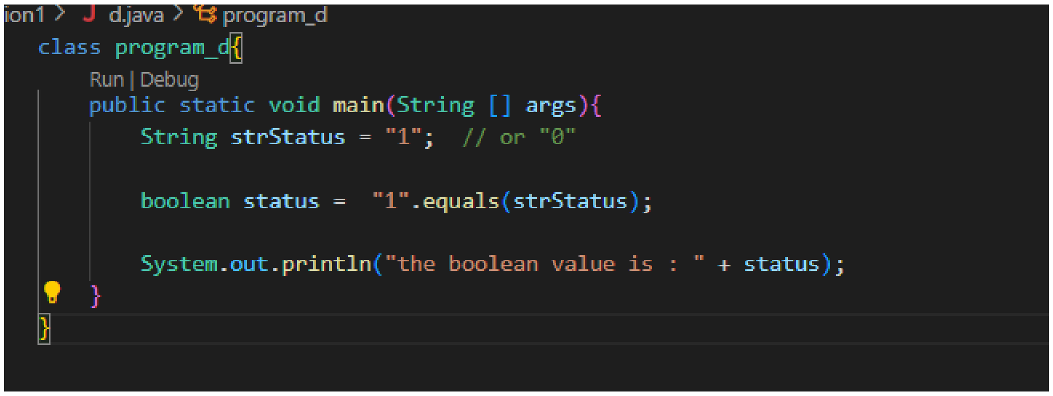


```
class program_c{
    public static void main(String [] args){
        String strStatus = "true";

        boolean boolstrstatus =  Boolean.parseBoolean(strStatus);

        System.out.println("the boolean representation of the string is : " +
        boolstrstatus);
    }
}
```

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"`).

Ans.

```
class program_d{

    public static void main(String [] args){

        String strStatus = "1";  // or "0"


        boolean status =  "1".equals(strStatus);


        System.out.println("the boolean value is : " + status);

    }

}
```

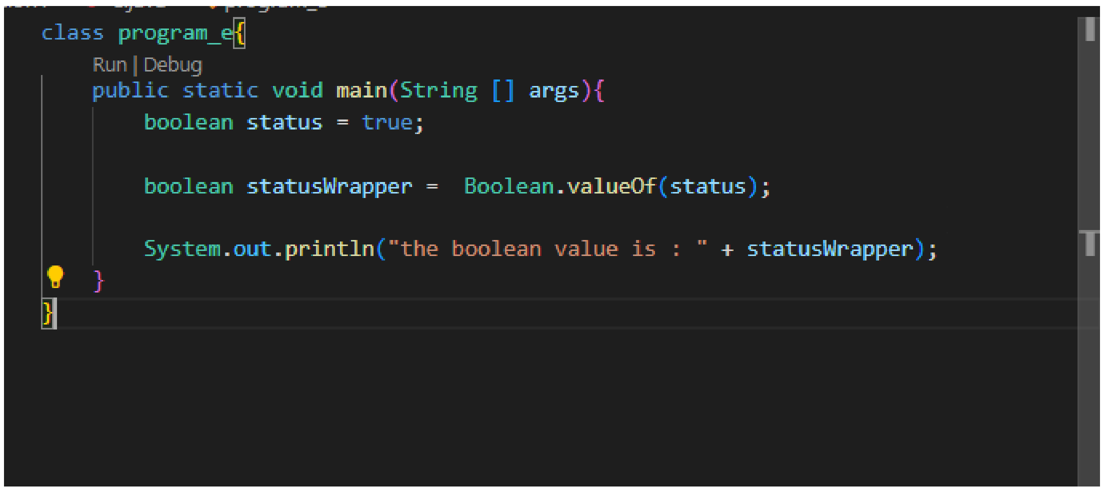


```java
class program_d{
    public static void main(String [] args){
        String strStatus = "1";   // or "0"

        boolean status =  "1".equals(strStatus);

        System.out.println("the boolean value is : " + status);
    }
}
```

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)`).

Ans. class program_e{

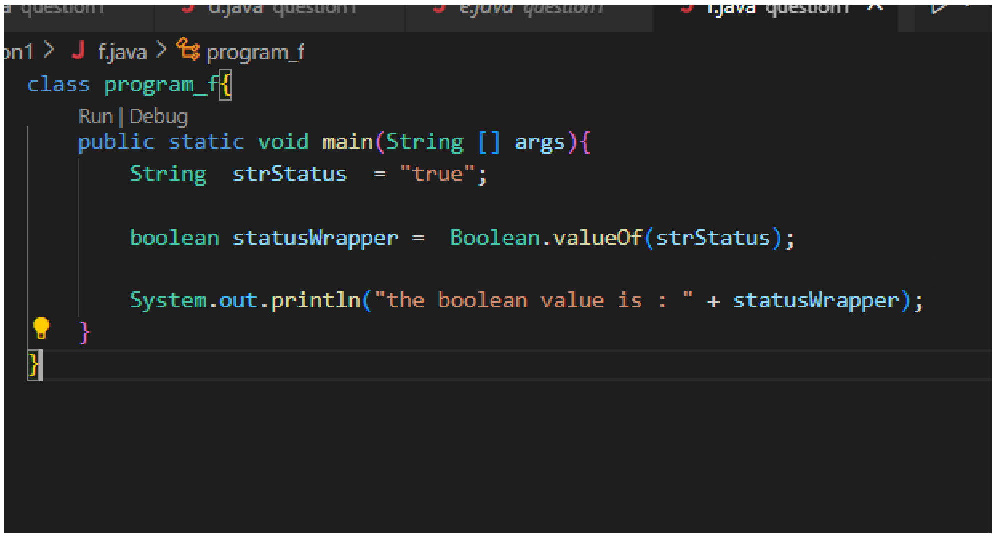
```
  public static void main(String [] args){

    boolean status = true;



    boolean statusWrapper =  Boolean.valueOf(status);



    System.out.println("the boolean value is : " + statusWrapper);

  }

}
```



```java
class program_e{
    public static void main(String [] args){
        boolean status = true;

        boolean statusWrapper =  Boolean.valueOf(status);

        System.out.println("the boolean value is : " + statusWrapper);
    }
}
```

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)`).

Ans.
```
class program_f{

  public static void main(String [] args){

    String  strStatus  = "true";



    boolean statusWrapper =  Boolean.valueOf(strStatus);



    System.out.println("the boolean value is : " + statusWrapper);

  }

}
```



a question d.java question1 e.java question1 f.java question1

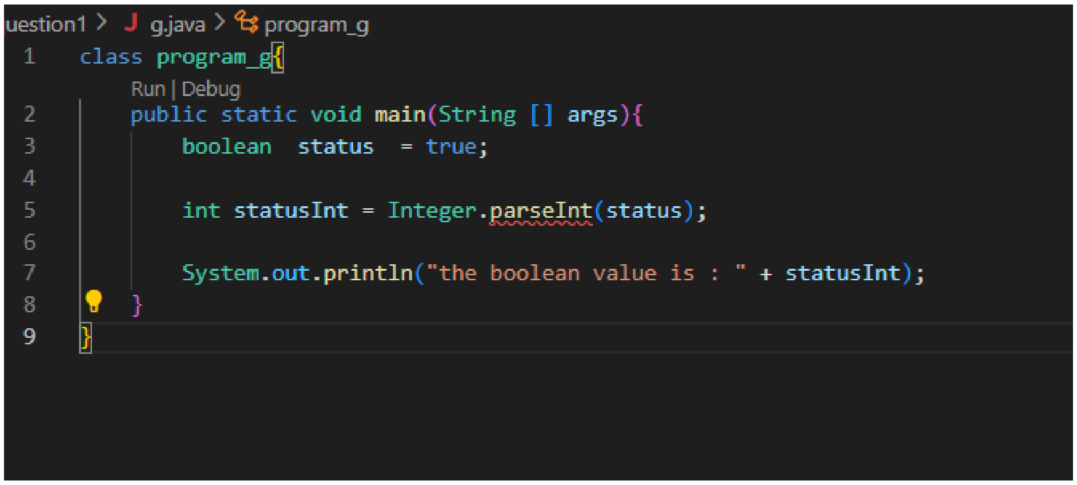
on1 > f.java > program_f

```
class program_f{
    Run | Debug
    public static void main(String [] args){
        String  strStatus  = "true";

        boolean statusWrapper =  Boolean.valueOf(strStatus);

        System.out.println("the boolean value is : " + statusWrapper);
    }
}
```

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

Ans.
```
class program_g{
          public static void main(String [] args){
            boolean  status  = true;

            int statusInt = Integer.parseInt(status);
```

```
        System.out.println("the boolean value is : " + statusInt);
    }
}
```



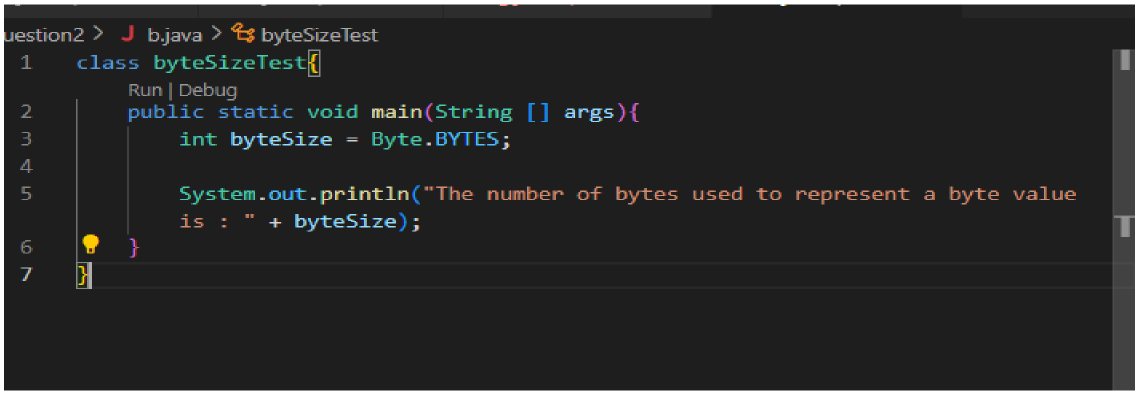
```
uestion1 > g.java > program_g
class program_g{
    Run | Debug
    public static void main(String [] args){
        boolean  status  = true;

        int statusInt = Integer.parseInt(status);

        System.out.println("the boolean value is : " + statusInt);
    }
}
```

2. Working with `java.lang.Byte`

b. Write a program to test how many bytes are used to represent a `byte` value using the `BYTES` field. (Hint: Use `Byte.BYTES`).

Ans. class byteSizeTest{

```
public static void main(String [] args){

    int byteSize = Byte.BYTES;



    System.out.println("The number of bytes used to represent a byte value is : " + byteSize);

}

}
```

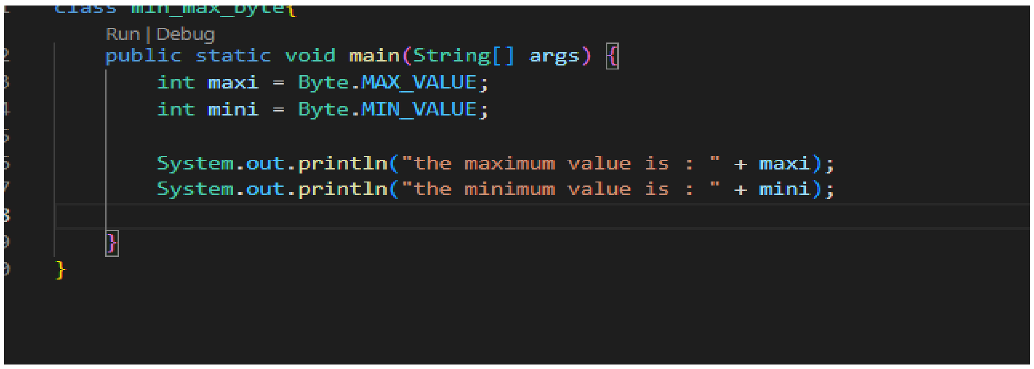


```
uestion2 > b.java > byteSizeTest
class byteSizeTest{
    Run | Debug
    public static void main(String [] args){
        int byteSize = Byte.BYTES;

        System.out.println("The number of bytes used to represent a byte value
        is : " + byteSize);
    }
}
```

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`).

Ans. class min_max_byte{

```
    public static void main(String[] args) {

        int maxi = Byte.MAX_VALUE;

        int mini = Byte.MIN_VALUE;



        System.out.println("the maximum value is : " + maxi);

        System.out.println("the minimum value is : " + mini);



    }

}
```

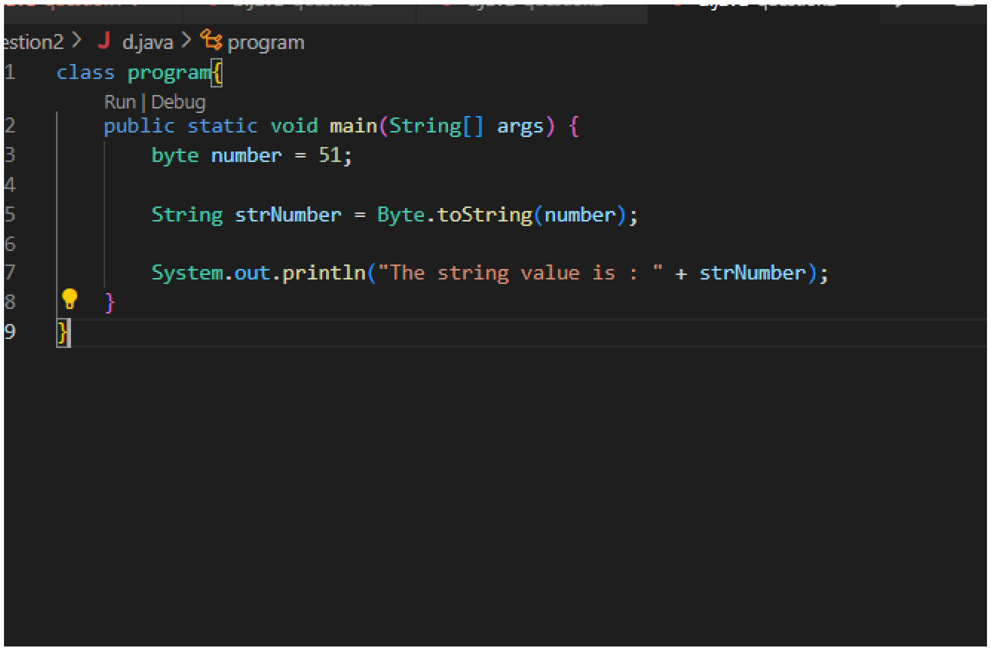


```
class min_max_byte{
    Run | Debug
    public static void main(String[] args) {
        int maxi = Byte.MAX_VALUE;
        int mini = Byte.MIN_VALUE;

        System.out.println("the maximum value is : " + maxi);
        System.out.println("the minimum value is : " + mini);

    }
}
```

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)`).

Ans. class program{

```
    public static void main(String[] args) {

        byte number = 51;



        String strNumber = Byte.toString(number);
    System.out.println("The string value is : " + strNumber);
  }
}
```

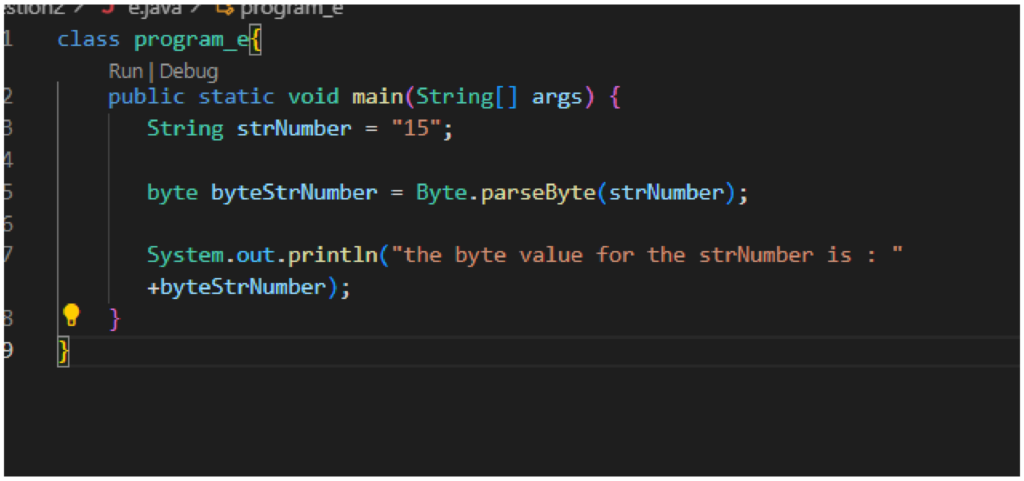


```
estion2 > d.java > program
class program{
    Run | Debug
    public static void main(String[] args) {
        byte number = 51;

        String strNumber = Byte.toString(number);

        System.out.println("The string value is :  " + strNumber);
    }
}
```

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)`).

Ans. class program_e{

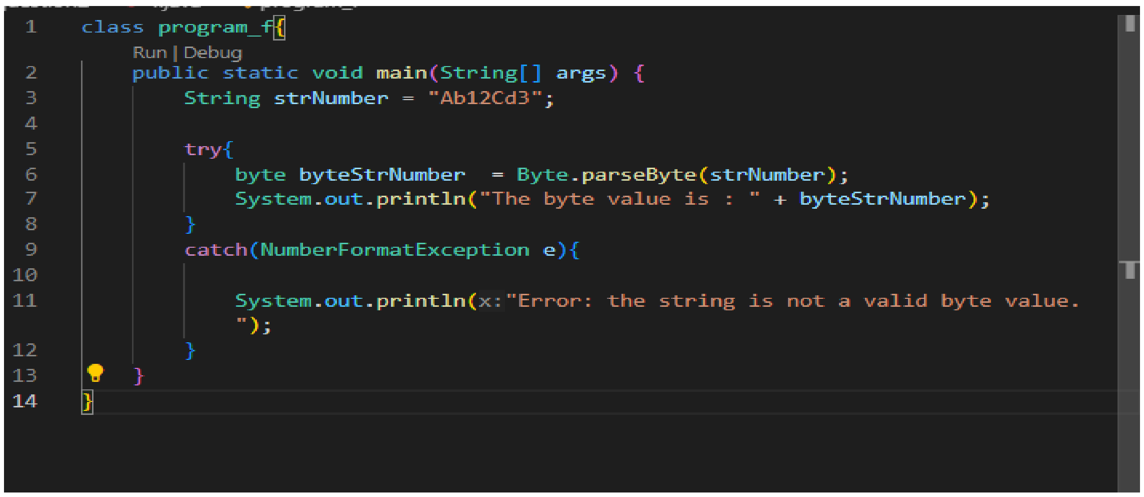
```
  public static void main(String[] args) {
    String strNumber = "15";


    byte byteStrNumber = Byte.parseByte(strNumber);


    System.out.println("the byte value for the strNumber is : "+byteStrNumber);
  }
}
```


```
stion2 > e.java > program_e
class program_e{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "15";

        byte byteStrNumber = Byte.parseByte(strNumber);

        System.out.println("the byte value for the strNumber is : "
        +byteStrNumber);
    }
}
```

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`).

Ans. class program_f{

```
  public static void main(String[] args) {

    String strNumber = "Ab12Cd3";



    try{

      byte byteStrNumber  = Byte.parseByte(strNumber);

      System.out.println("The byte value is : " + byteStrNumber);

    }

    catch(NumberFormatException e){



      System.out.println("Error: the string is not a valid byte value.");

    }

  }

}
```

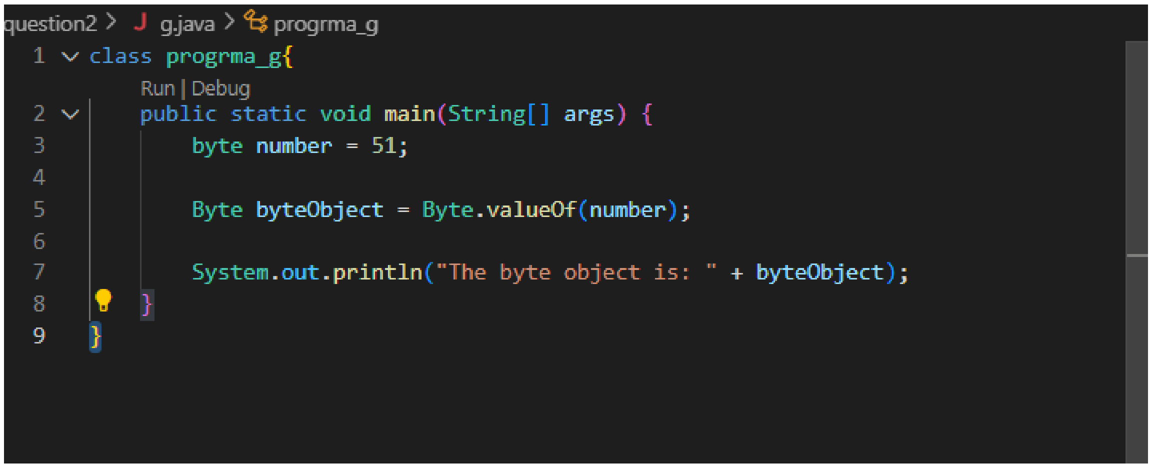


```
class program_f{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "Ab12Cd3";

        try{
            byte byteStrNumber  = Byte.parseByte(strNumber);
            System.out.println("The byte value is : " + byteStrNumber);
        }
        catch(NumberFormatException e){

            System.out.println(x:"Error: the string is not a valid byte value.
            ");
        }
    }
}
```

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)`).

Ans. class progrma_g{

```
  public static void main(String[] args) {

    byte number = 51;



    Byte byteObject = Byte.valueOf(number);



    System.out.println("The byte object is: " + byteObject);

  }

}
```



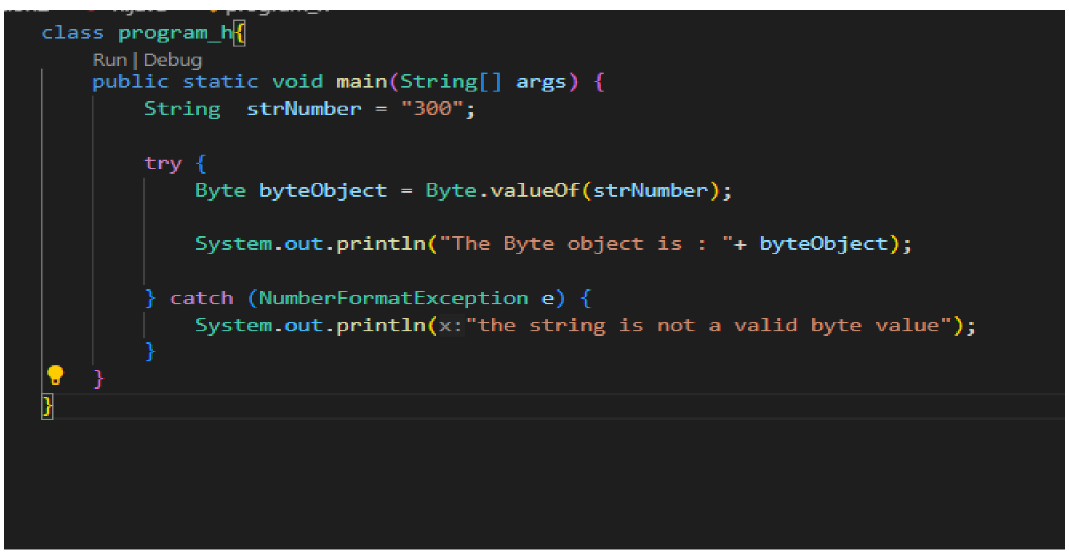
question2 > g.java > progrma_g

```
class progrma_g{
    Run | Debug
    public static void main(String[] args) {
        byte number = 51;

        Byte byteObject = Byte.valueOf(number);

        System.out.println("The byte object is: " + byteObject);
    }
}
```

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)`).

Ans.

```
class program_h{

   public static void main(String[] args) {

      String  strNumber = "300";



      try {

         Byte byteObject = Byte.valueOf(strNumber);



         System.out.println("The Byte object is : "+ byteObject);



      } catch (NumberFormatException e) {

         System.out.println("the string is not a valid byte value");

      }

   }

}
```



```
class program_h{
    Run | Debug
    public static void main(String[] args) {
        String  strNumber = "300";

        try {
            Byte byteObject = Byte.valueOf(strNumber);

            System.out.println("The Byte object is : "+ byteObject);

        } catch (NumberFormatException e) {
            System.out.println(x:"the string is not a valid byte value");
        }
    }
}
```

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`).

Ans. package question3;

class program_b{

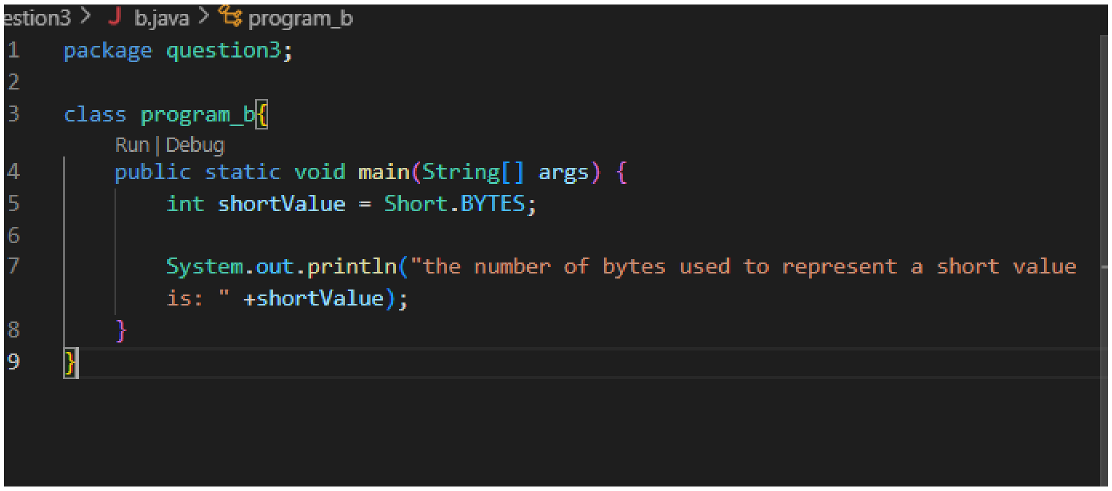
  public static void main(String[] args) {

    int shortValue = Short.BYTES;

    System.out.println("the number of bytes used to represent a short value is: " +shortValue);

  }

}



```
package question3;

class program_b{
    public static void main(String[] args) {
        int shortValue = Short.BYTES;

        System.out.println("the number of bytes used to represent a short value
        is: " +shortValue);
    }
}
```

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`).

Ans.

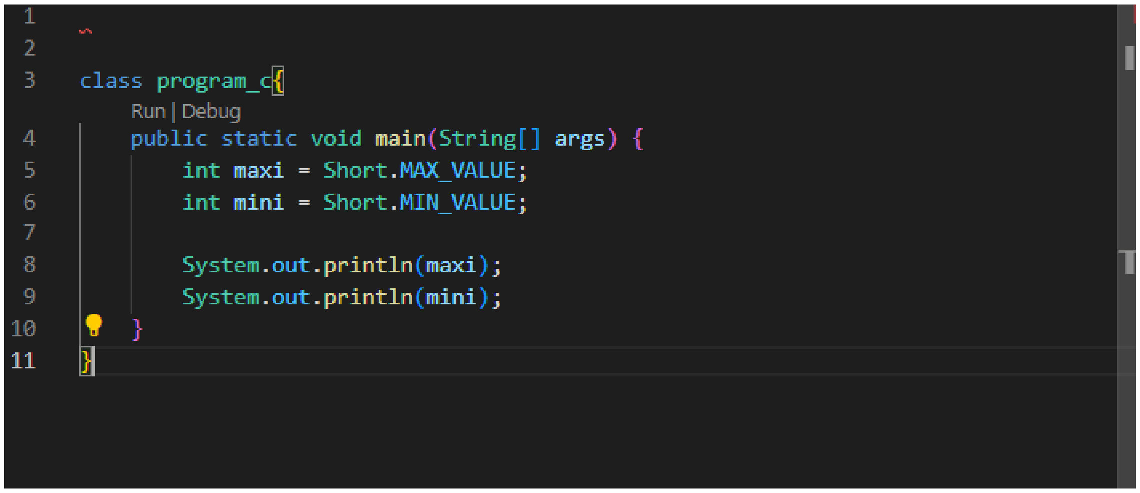
class program_c{

  public static void main(String[] args) {

    int maxi = Short.MAX_VALUE;

    int mini = Short.MIN_VALUE;

```
        System.out.println(maxi);

        System.out.println(mini);

    }

}
```

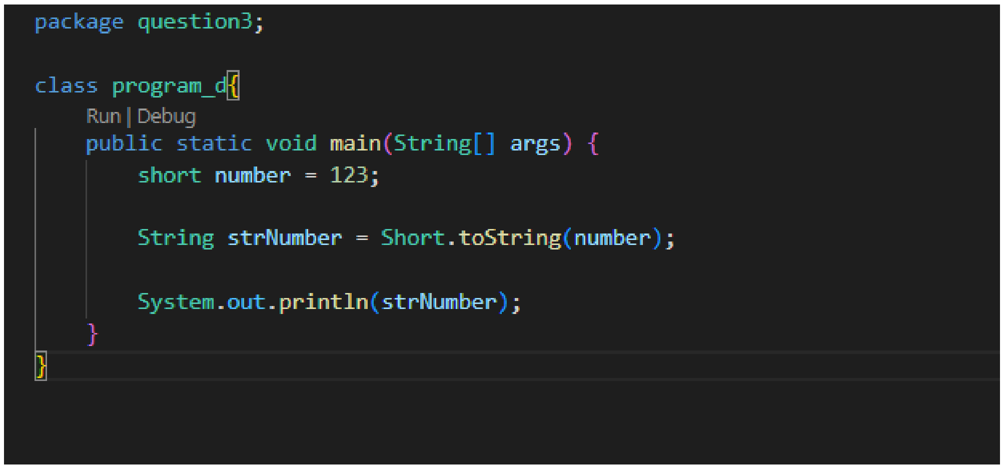


```
class program_c{
    Run | Debug
    public static void main(String[] args) {
        int maxi = Short.MAX_VALUE;
        int mini = Short.MIN_VALUE;

        System.out.println(maxi);
        System.out.println(mini);
    }
}
```

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)`).

Ans. package question3;

```
class program_d{

    public static void main(String[] args) {

        short number = 123;



        String strNumber = Short.toString(number);



        System.out.println(strNumber);

    }

}
```

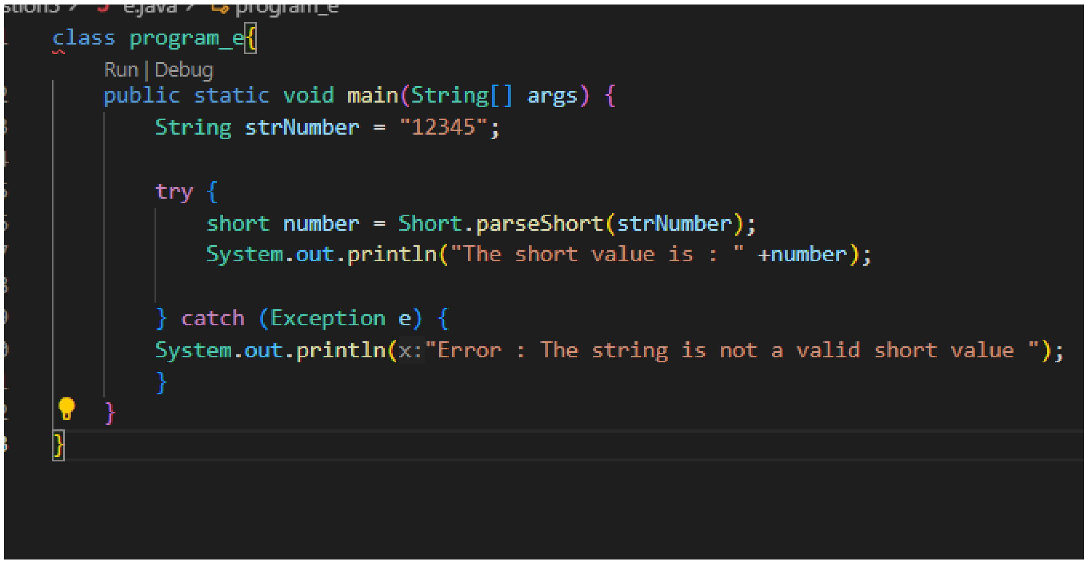


```
package question3;

class program_d{
    Run | Debug
    public static void main(String[] args) {
        short number = 123;

        String strNumber = Short.toString(number);

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

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)`).

Ans. class program_e{

```
  public static void main(String[] args) {

    String strNumber = "12345";



    try {

      short number = Short.parseShort(strNumber);

      System.out.println("The short value is : " +number);



    } catch (Exception e) {

    System.out.println("Error : The string is not a valid short value ");

    }

  }

}
```



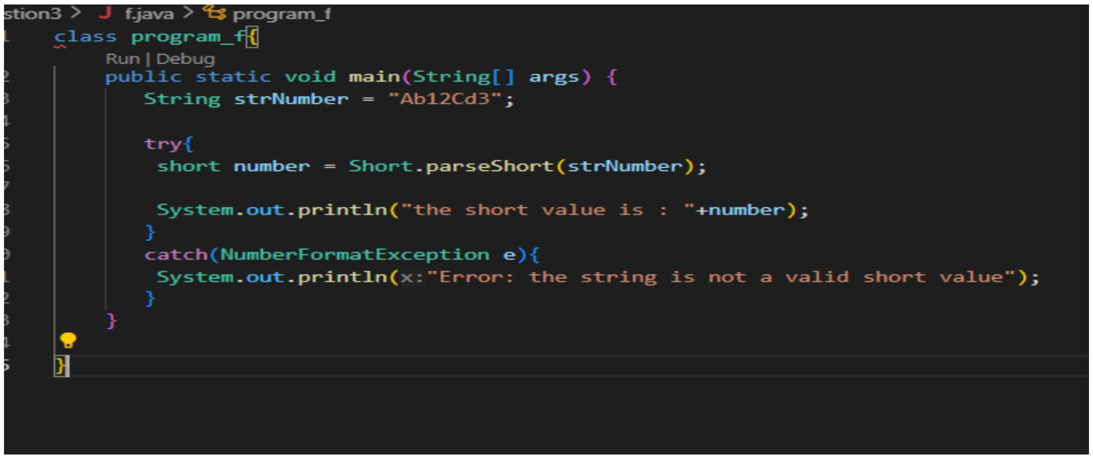
```
class program_e{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "12345";

        try {
            short number = Short.parseShort(strNumber);
            System.out.println("The short value is : " +number);

        } catch (Exception e) {
        System.out.println(x:"Error : The string is not a valid short value ");
        }
    }
}
```

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`).

Ans. class program_f{

```
  public static void main(String[] args) {

    String strNumber = "Ab12Cd3";



    try{

     short number = Short.parseShort(strNumber);



     System.out.println("the short value is : "+number);

    }

    catch(NumberFormatException e){

     System.out.println("Error: the string is not a valid short value");

    }

  }
```

}



```
stion3 > f.java > program_f
class program_f{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "Ab12Cd3";

        try{
         short number = Short.parseShort(strNumber);

         System.out.println("the short value is : "+number);
        }
        catch(NumberFormatException e){
         System.out.println(x:"Error: the string is not a valid short value");
        }
    }

}
```

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)`).

Ans. class program_g{

public static void main(String[] args) {

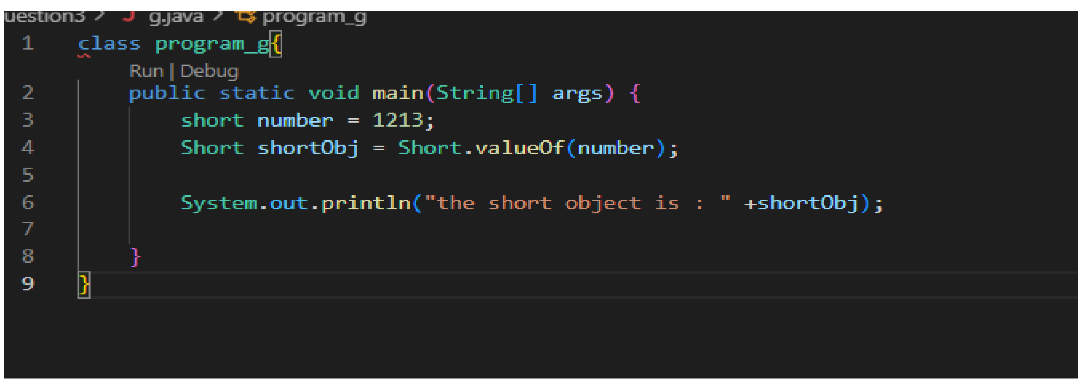
short number = 1213;

Short shortObj = Short.valueOf(number);

System.out.println("the short object is : " +shortObj);

}

}

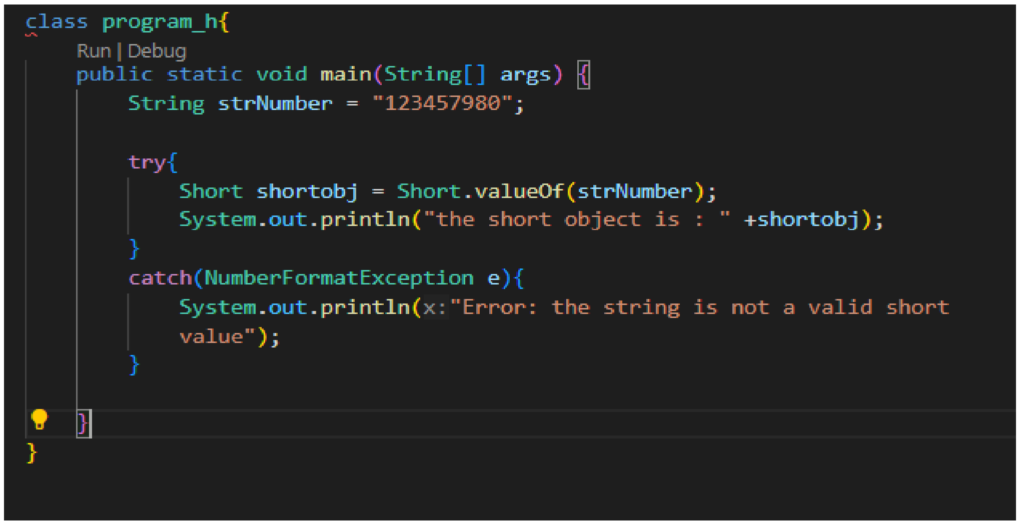


```
uestion3 > g.java > program_g
class program_g{
    Run | Debug
    public static void main(String[] args) {
        short number = 1213;
        Short shortObj = Short.valueOf(number);

        System.out.println("the short object is : " +shortObj);

    }
}
```

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)`).

Ans. class program_h{

```
  public static void main(String[] args) {

    String strNumber = "123457980";



    try{

      Short shortobj = Short.valueOf(strNumber);

      System.out.println("the short object is : " +shortobj);

    }

    catch(NumberFormatException e){

      System.out.println("Error: the string is not a valid short value");

    }



  }

}
```



```
class program_h{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "123457980";

        try{
            Short shortobj = Short.valueOf(strNumber);
            System.out.println("the short object is : " +shortobj);
        }
        catch(NumberFormatException e){
            System.out.println(x:"Error: the string is not a valid short 
            value");
        }

    }
}
```

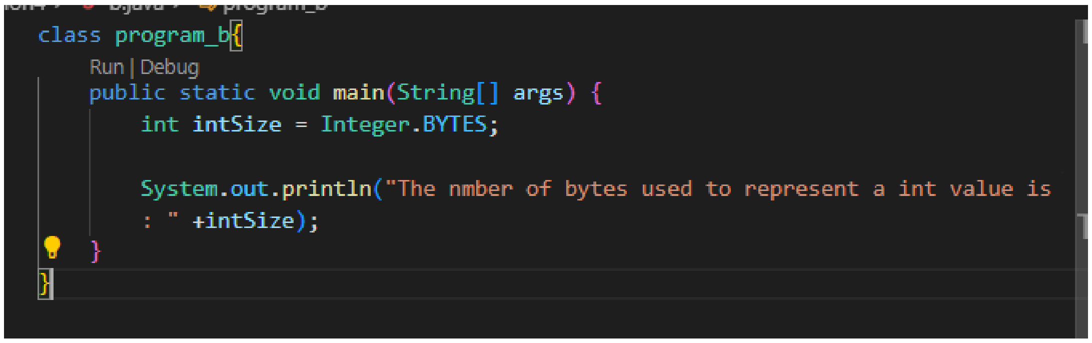
4. Working with `java.lang.Integer`

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`).

Ans. class program_b{

```
  public static void main(String[] args) {

    int intSize = Integer.BYTES;



    System.out.println("The nmber of bytes used to represent a int value is : " +intSize);

  }

}
```



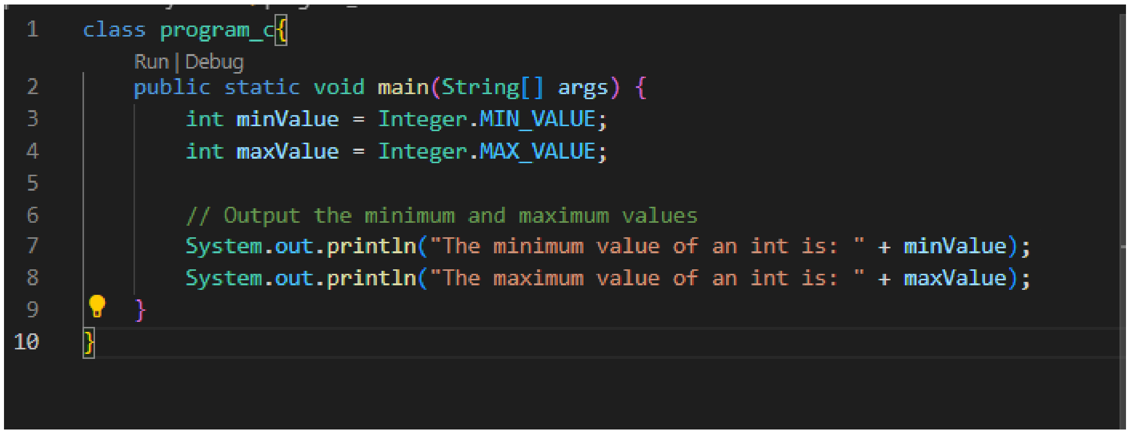
```
class program_b{
    Run | Debug
    public static void main(String[] args) {
        int intSize = Integer.BYTES;

        System.out.println("The nmber of bytes used to represent a int value is
        : " +intSize);
    }
}
```

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`).

Ans. class program_c{

```
  public static void main(String[] args) {

    int minValue = Integer.MIN_VALUE;

    int maxValue = Integer.MAX_VALUE;



    // Output the minimum and maximum values

    System.out.println("The minimum value of an int is: " + minValue);
```

```
        System.out.println("The maximum value of an int is: " + maxValue);

    }

}
```

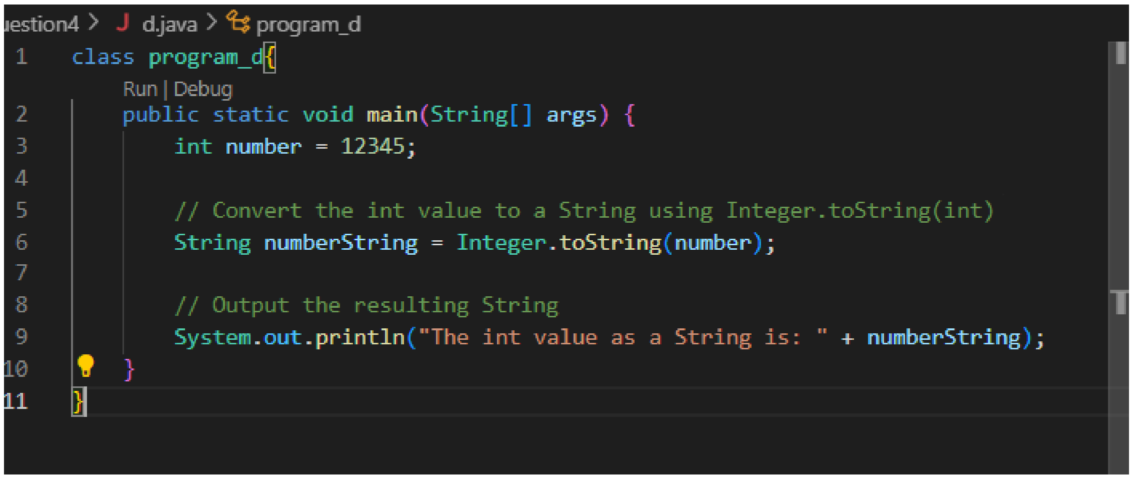


```
class program_c{
    Run | Debug
    public static void main(String[] args) {
        int minValue = Integer.MIN_VALUE;
        int maxValue = Integer.MAX_VALUE;

        // Output the minimum and maximum values
        System.out.println("The minimum value of an int is: " + minValue);
        System.out.println("The maximum value of an int is: " + maxValue);
    }
}
```

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)`).

Ans. class program_d{

```
    public static void main(String[] args) {

        int number = 12345;



        // Convert the int value to a String using Integer.toString(int)

        String numberString = Integer.toString(number);



        // Output the resulting String

        System.out.println("The int value as a String is: " + numberString);

    }

}
```

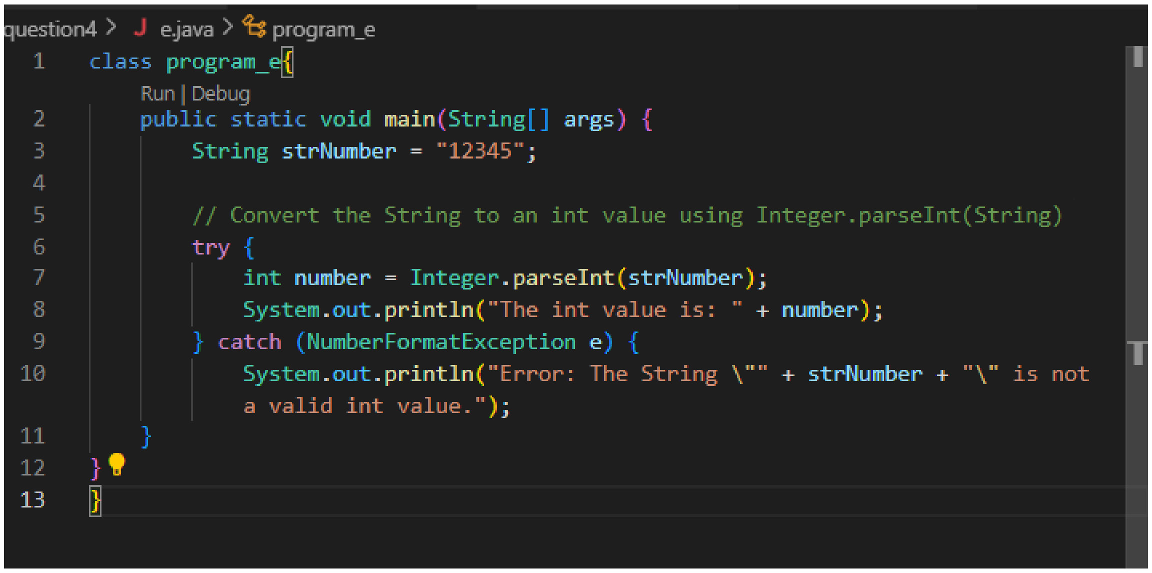


```java
class program_d{
    public static void main(String[] args) {
        int number = 12345;

        // Convert the int value to a String using Integer.toString(int)
        String numberString = Integer.toString(number);

        // Output the resulting String
        System.out.println("The int value as a String is: " + numberString);
    }
}
```

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)`).

Ans. class program_e{

```java
  public static void main(String[] args) {
    String strNumber = "12345";


    // Convert the String to an int value using Integer.parseInt(String)
    try {
      int number = Integer.parseInt(strNumber);
      System.out.println("The int value is: " + number);
    } catch (NumberFormatException e) {
      System.out.println("Error: The String \"" + strNumber + "\" is not a valid int value.");
  }
}
}
```

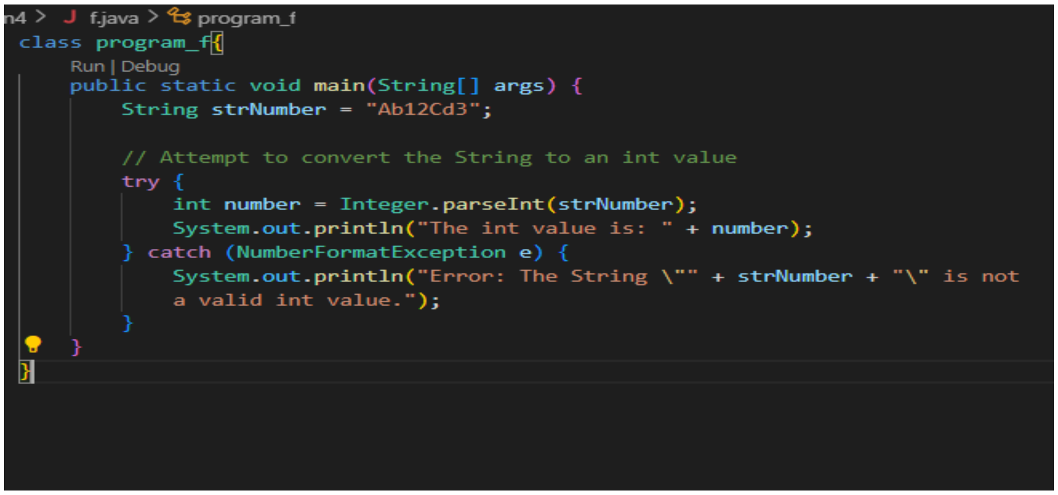


```java
class program_e{
    public static void main(String[] args) {
        String strNumber = "12345";

        // Convert the String to an int value using Integer.parseInt(String)
        try {
            int number = Integer.parseInt(strNumber);
            System.out.println("The int value is: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid int value.");
        }
    }
}
```

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`).

Ans. class program_f{

```java
  public static void main(String[] args) {

    String strNumber = "Ab12Cd3";



    // Attempt to convert the String to an int value

    try {

      int number = Integer.parseInt(strNumber);

      System.out.println("The int value is: " + number);

    } catch (NumberFormatException e) {

      System.out.println("Error: The String \"" + strNumber + "\" is not a valid int value.");

    }

  }

}
```


```
n4 > J f.java > program_f
class program_f{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "Ab12Cd3";

        // Attempt to convert the String to an int value
        try {
            int number = Integer.parseInt(strNumber);
            System.out.println("The int value is: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid int value.");
        }
    }
}
```

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)`).

Ans. class program_g{

public static void main(String[] args) {

int number = 12345;

// Convert the int value to an Integer object using Integer.valueOf(int)

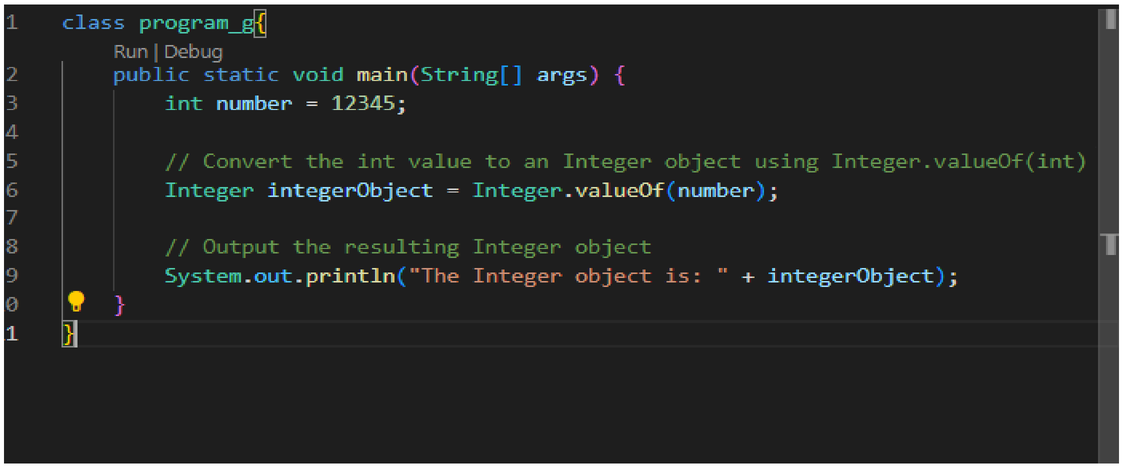
Integer integerObject = Integer.valueOf(number);

// Output the resulting Integer object

System.out.println("The Integer object is: " + integerObject);

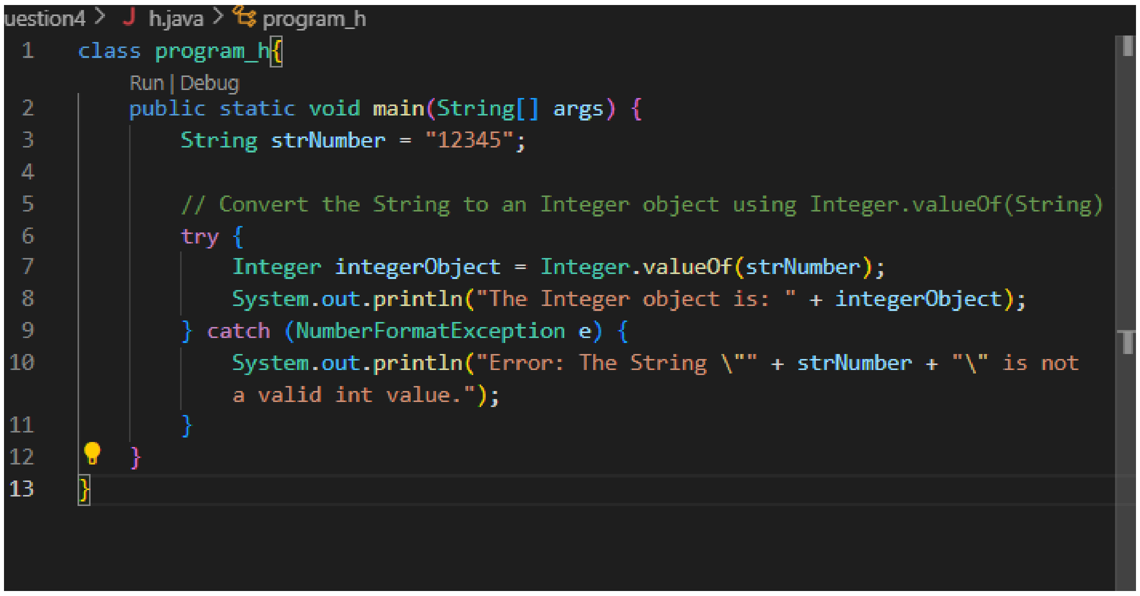
}

}


```
class program_g{
    Run | Debug
    public static void main(String[] args) {
        int number = 12345;

        // Convert the int value to an Integer object using Integer.valueOf(int)
        Integer integerObject = Integer.valueOf(number);

        // Output the resulting Integer object
        System.out.println("The Integer object is: " + integerObject);
    }
}
```

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)`).

Ans.
```
class program_h{

  public static void main(String[] args) {

    String strNumber = "12345";



    // Convert the String to an Integer object using Integer.valueOf(String)

    try {

      Integer integerObject = Integer.valueOf(strNumber);

      System.out.println("The Integer object is: " + integerObject);

    } catch (NumberFormatException e) {

      System.out.println("Error: The String \"" + strNumber + "\" is not a valid int value.");

    }

  }

}
```



uestion4 > h.java > program_h

```
class program_h{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "12345";

        // Convert the String to an Integer object using Integer.valueOf(String)
        try {
            Integer integerObject = Integer.valueOf(strNumber);
            System.out.println("The Integer object is: " + integerObject);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid int value.");
        }
    }
}
```

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)`).

Ans. class program_i{

  public static void main(String[] args) {

    int a = 10;

    int b = 20;

    // Add the two integers using Integer.sum(int, int)

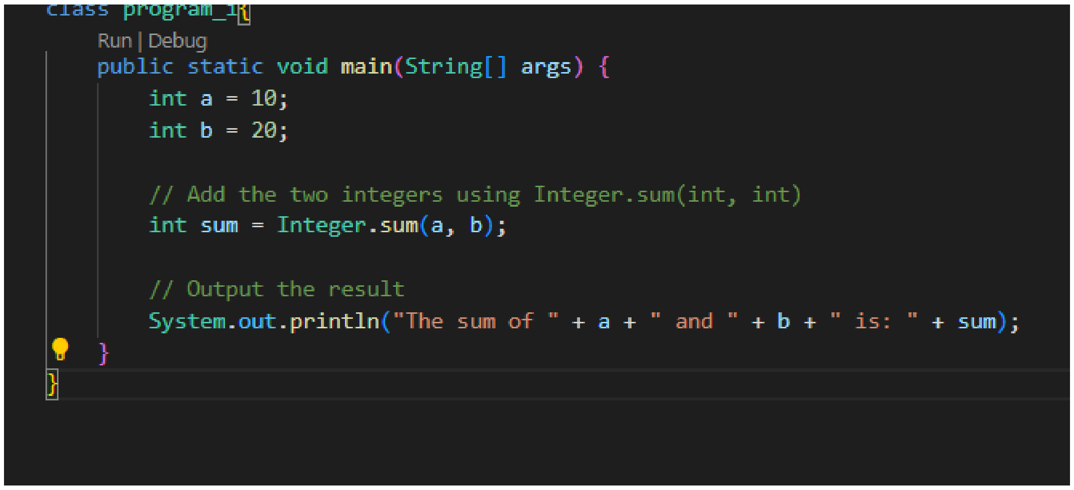
    int sum = Integer.sum(a, b);

    // Output the result

    System.out.println("The sum of " + a + " and " + b + " is: " + sum);

  }

}

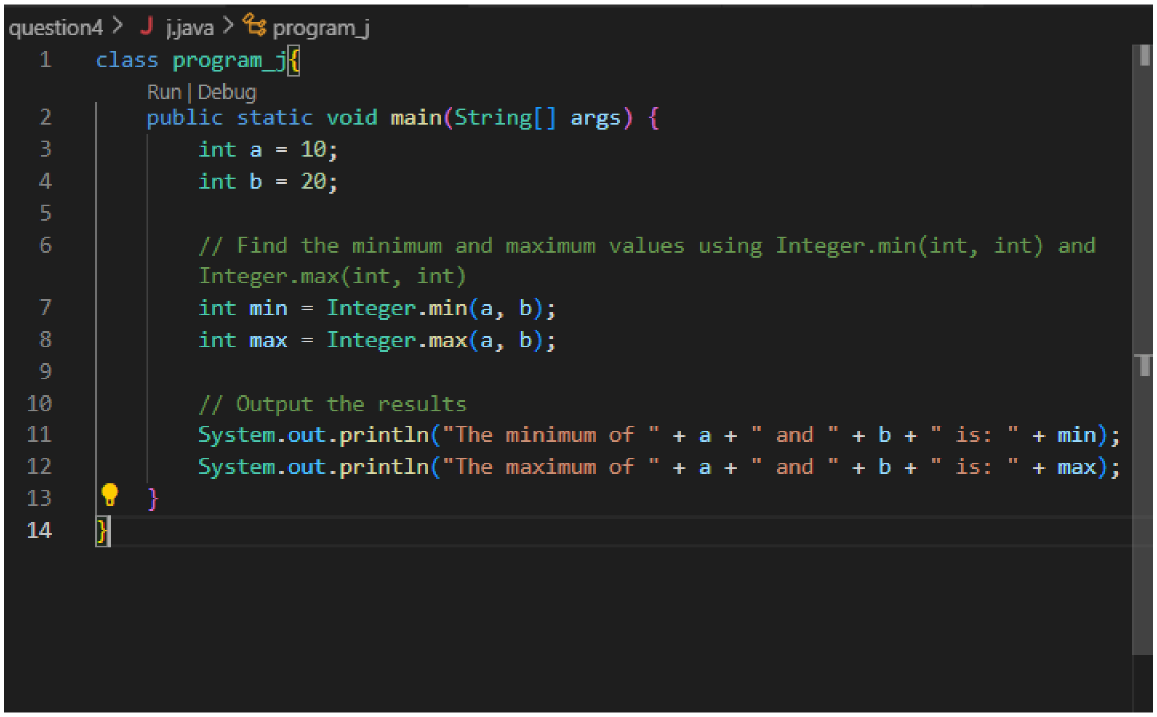


```
class program_i{
    Run | Debug
    public static void main(String[] args) {
        int a = 10;
        int b = 20;

        // Add the two integers using Integer.sum(int, int)
        int sum = Integer.sum(a, b);

        // Output the result
        System.out.println("The sum of " + a + " and " + b + " is: " + 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)`).

Ans. class program_j{

```
    public static void main(String[] args) {

        int a = 10;

        int b = 20;



        // Find the minimum and maximum values using Integer.min(int, int) and Integer.max(int, int)

        int min = Integer.min(a, b);

        int max = Integer.max(a, b);



        // Output the results

        System.out.println("The minimum of " + a + " and " + b + " is: " + min);

        System.out.println("The maximum of " + a + " and " + b + " is: " + max);

    }

}
```

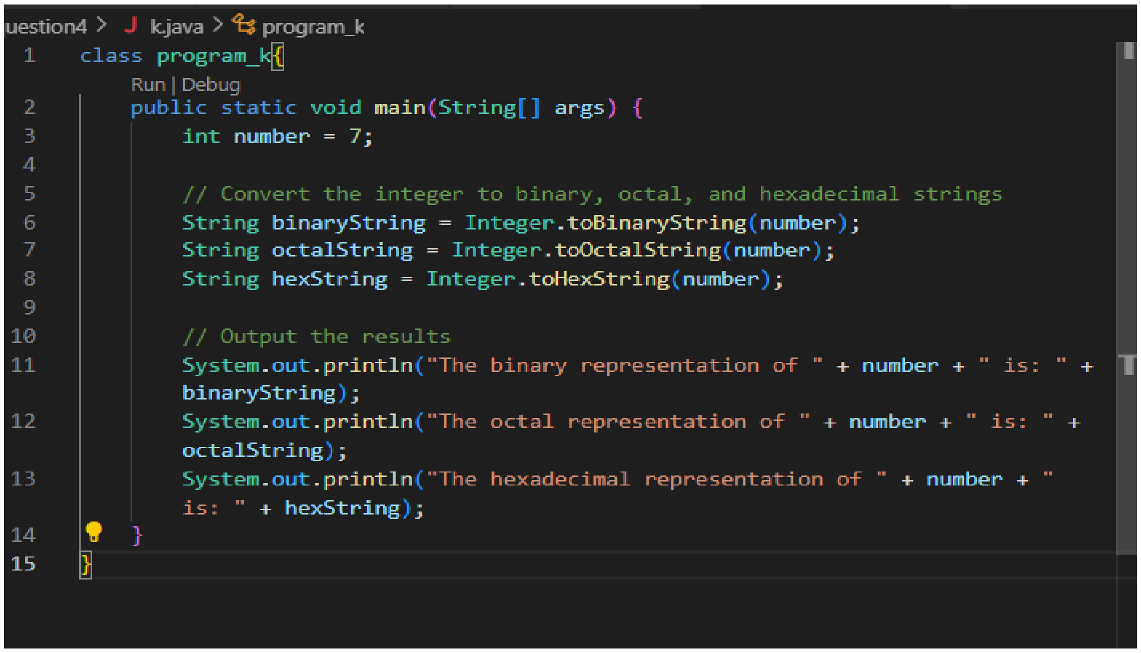


```
class program_j{
    public static void main(String[] args) {
        int a = 10;
        int b = 20;

        // Find the minimum and maximum values using Integer.min(int, int) and Integer.max(int, int)
        int min = Integer.min(a, b);
        int max = Integer.max(a, b);

        // Output the results
        System.out.println("The minimum of " + a + " and " + b + " is: " + min);
        System.out.println("The maximum of " + a + " and " + b + " is: " + 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)`).

Ans. class program_k{

```
  public static void main(String[] args) {

    int number = 7;

    // Convert the integer to binary, octal, and hexadecimal strings

    String binaryString = Integer.toBinaryString(number);

    String octalString = Integer.toOctalString(number);

    String hexString = Integer.toHexString(number);

    // Output the results

    System.out.println("The binary representation of " + number + " is: " + binaryString);

    System.out.println("The octal representation of " + number + " is: " + octalString);

    System.out.println("The hexadecimal representation of " + number + " is: " + hexString);

  }

}
```



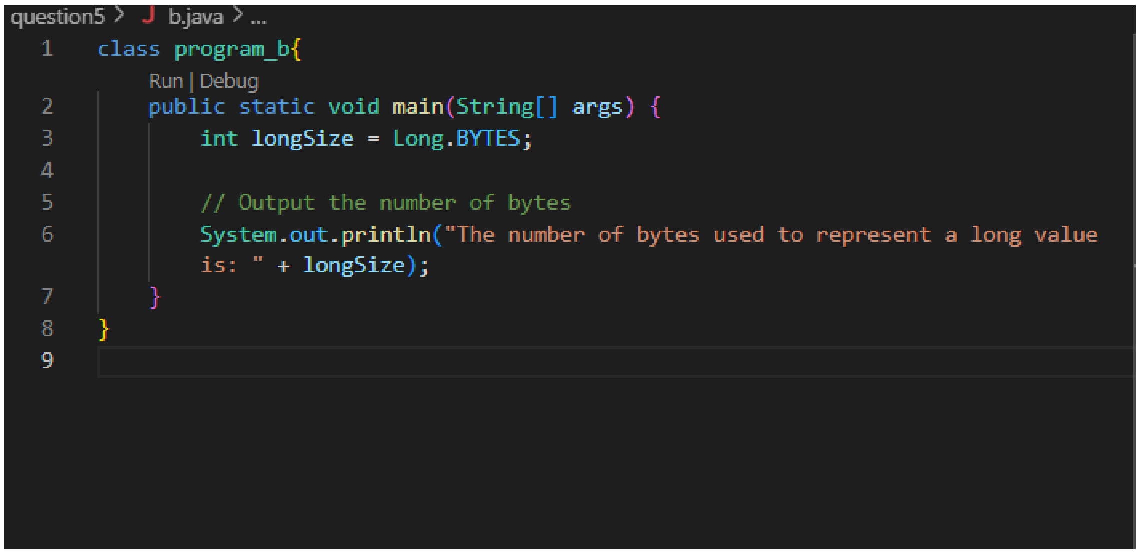
uestion4 > k.java > program_k

```
class program_k{
    public static void main(String[] args) {
        int number = 7;

        // Convert the integer to binary, octal, and hexadecimal strings
        String binaryString = Integer.toBinaryString(number);
        String octalString = Integer.toOctalString(number);
        String hexString = Integer.toHexString(number);

        // Output the results
        System.out.println("The binary representation of " + number + " is: " +
        binaryString);
        System.out.println("The octal representation of " + number + " is: " +
        octalString);
        System.out.println("The hexadecimal representation of " + number + "
        is: " + hexString);
    }
}
```

5. Working with `java.lang.Long`

b. Write a program to test how many bytes are used to represent a `long` value using the `BYTES` field. (Hint: Use `Long.BYTES`).

Ans.
```
class program_b{

  public static void main(String[] args) {

    int longSize = Long.BYTES;



    // Output the number of bytes

    System.out.println("The number of bytes used to represent a long value is: " + longSize);

  }

}
```



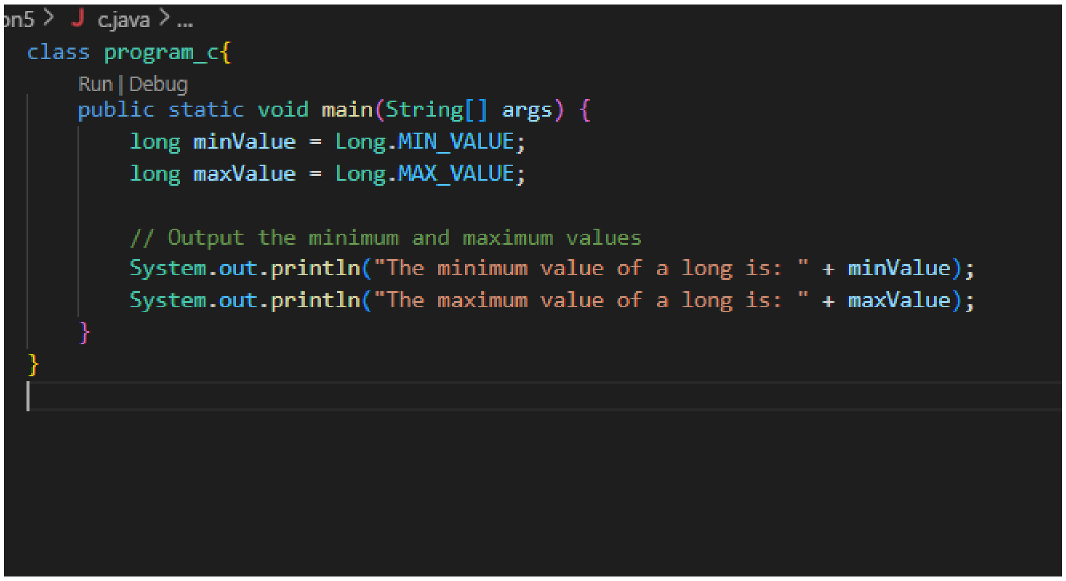
question5 > b.java > ...

```
class program_b{
    public static void main(String[] args) {
        int longSize = Long.BYTES;

        // Output the number of bytes
        System.out.println("The number of bytes used to represent a long value is: " + longSize);
    }
}

```

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`).

Ans.
```
class program_c{

  public static void main(String[] args) {

    long minValue = Long.MIN_VALUE;
```

```
        long maxValue = Long.MAX_VALUE;


        // Output the minimum and maximum values

        System.out.println("The minimum value of a long is: " + minValue);

        System.out.println("The maximum value of a long is: " + maxValue);

    }

}
```

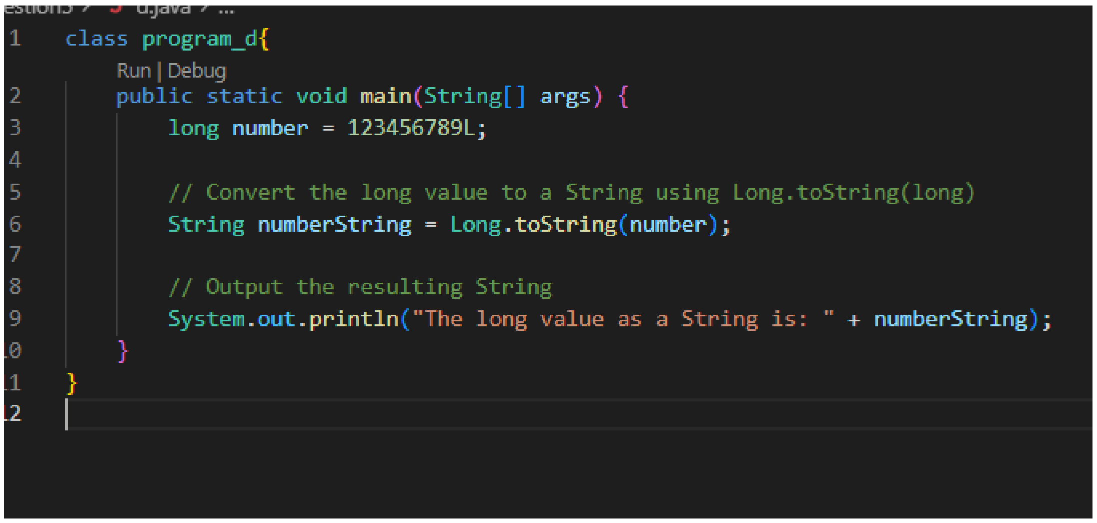


```
on5 > J c.java > ...
class program_c{
    Run | Debug
    public static void main(String[] args) {
        long minValue = Long.MIN_VALUE;
        long maxValue = Long.MAX_VALUE;

        // Output the minimum and maximum values
        System.out.println("The minimum value of a long is: " + minValue);
        System.out.println("The maximum value of a long is: " + maxValue);
    }
}
```

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)`).

Ans. class program_d{

```
    public static void main(String[] args) {

        long number = 123456789L;


        // Convert the long value to a String using Long.toString(long)

        String numberString = Long.toString(number);
        // Output the resulting String

        System.out.println("The long value as a String is: " + numberString);

    }

}
```

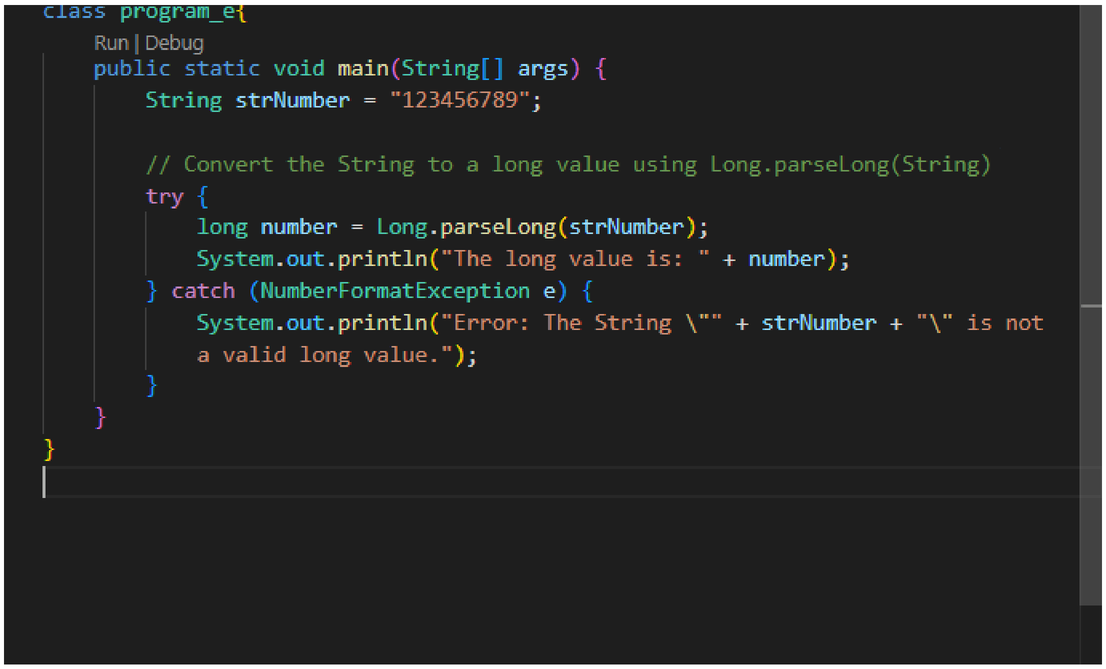


```
estions > J d.java > ...
class program_d{
    Run | Debug
    public static void main(String[] args) {
        long number = 123456789L;

        // Convert the long value to a String using Long.toString(long)
        String numberString = Long.toString(number);

        // Output the resulting String
        System.out.println("The long value as a String is: " + numberString);
    }
}
```

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)`).

Ans. class program_e{

```
    public static void main(String[] args) {

        String strNumber = "123456789";



        // Convert the String to a long value using Long.parseLong(String)

        try {

            long number = Long.parseLong(strNumber);

            System.out.println("The long value is: " + number);

        } catch (NumberFormatException e) {

            System.out.println("Error: The String \"" + strNumber + "\" is not a valid long value.");
    }

  }

}
```

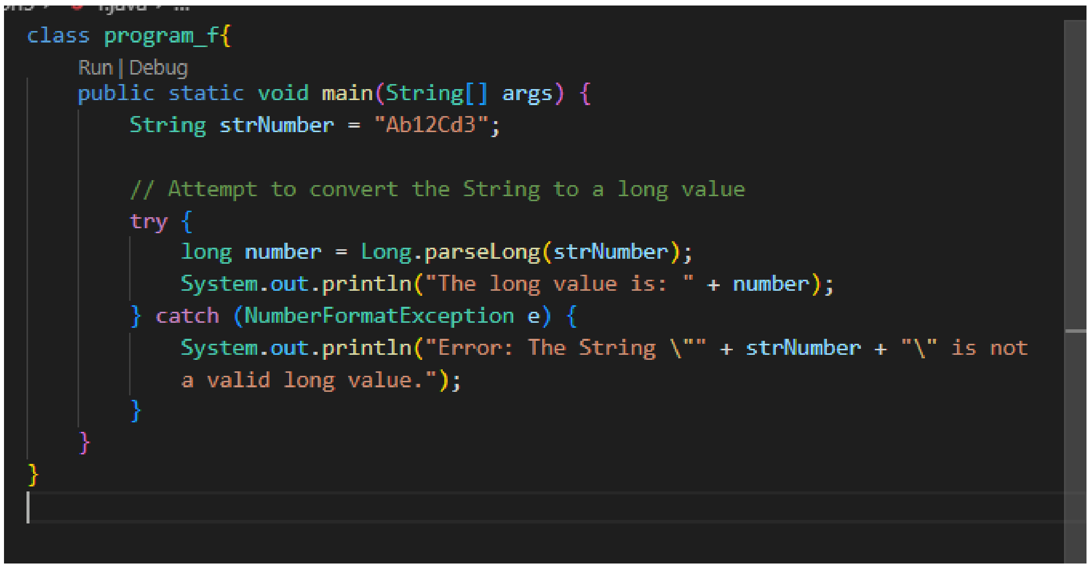


```
class program_e{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "123456789";

        // Convert the String to a long value using Long.parseLong(String)
        try {
            long number = Long.parseLong(strNumber);
            System.out.println("The long value is: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid long value.");
        }
    }
}
```

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`).

Ans. class program_f{

```
  public static void main(String[] args) {

    String strNumber = "Ab12Cd3";



    // Attempt to convert the String to a long value

    try {

      long number = Long.parseLong(strNumber);

      System.out.println("The long value is: " + number);

    } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not a valid long value.");

        }

    }

}
```



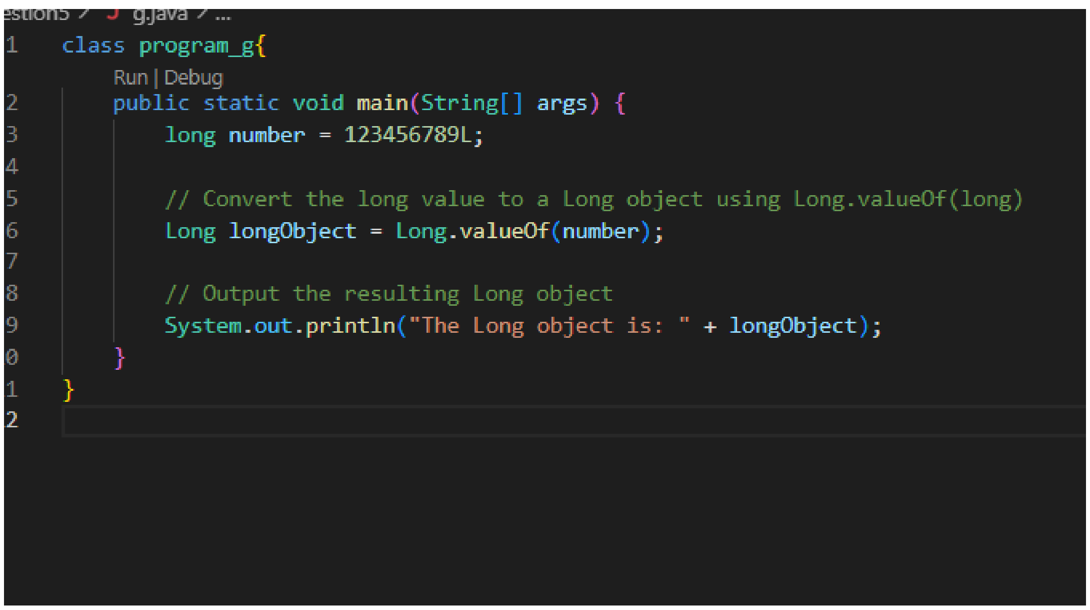
```
class program_f{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "Ab12Cd3";

        // Attempt to convert the String to a long value
        try {
            long number = Long.parseLong(strNumber);
            System.out.println("The long value is: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid long value.");
        }
    }
}
```

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)`).

Ans. class program_g{

```
    public static void main(String[] args) {

        long number = 123456789L;



        // Convert the long value to a Long object using Long.valueOf(long)

        Long longObject = Long.valueOf(number);



        // Output the resulting Long object

        System.out.println("The Long object is: " + longObject);

    }
```

}



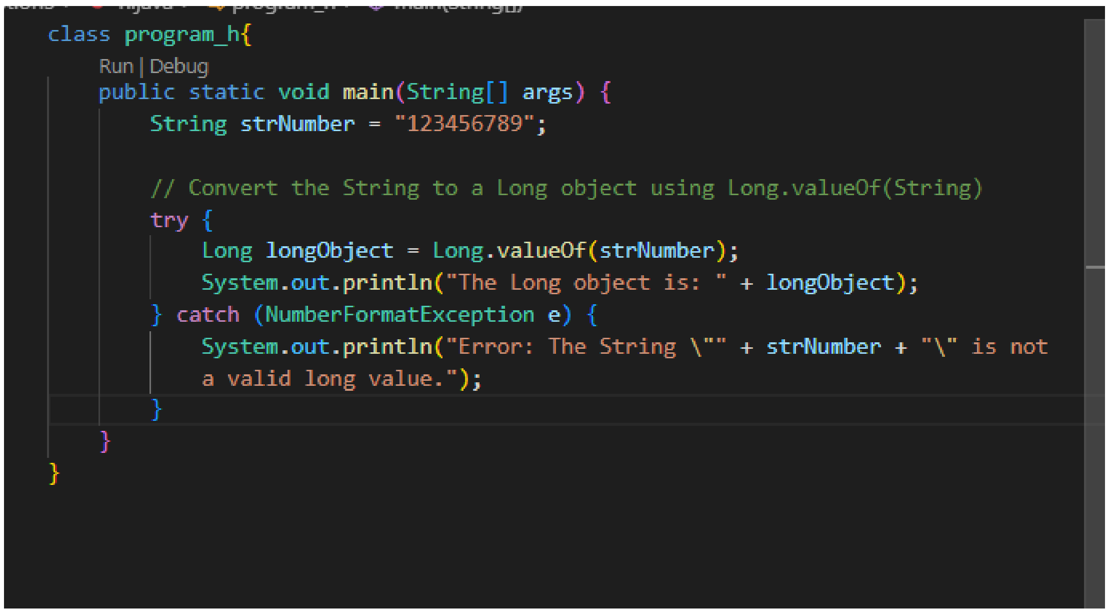
```
estion5 > J g.java > ...
class program_g{
    Run | Debug
    public static void main(String[] args) {
        long number = 123456789L;

        // Convert the long value to a Long object using Long.valueOf(long)
        Long longObject = Long.valueOf(number);

        // Output the resulting Long object
        System.out.println("The Long object is: " + longObject);
    }
}
```

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)`).

Ans. class program_h{

```
  public static void main(String[] args) {

    String strNumber = "123456789";



    // Convert the String to a Long object using Long.valueOf(String)

    try {

      Long longObject = Long.valueOf(strNumber);

      System.out.println("The Long object is: " + longObject);

    } catch (NumberFormatException e) {

      System.out.println("Error: The String \"" + strNumber + "\" is not a valid long value.");

    }

  }
```

}



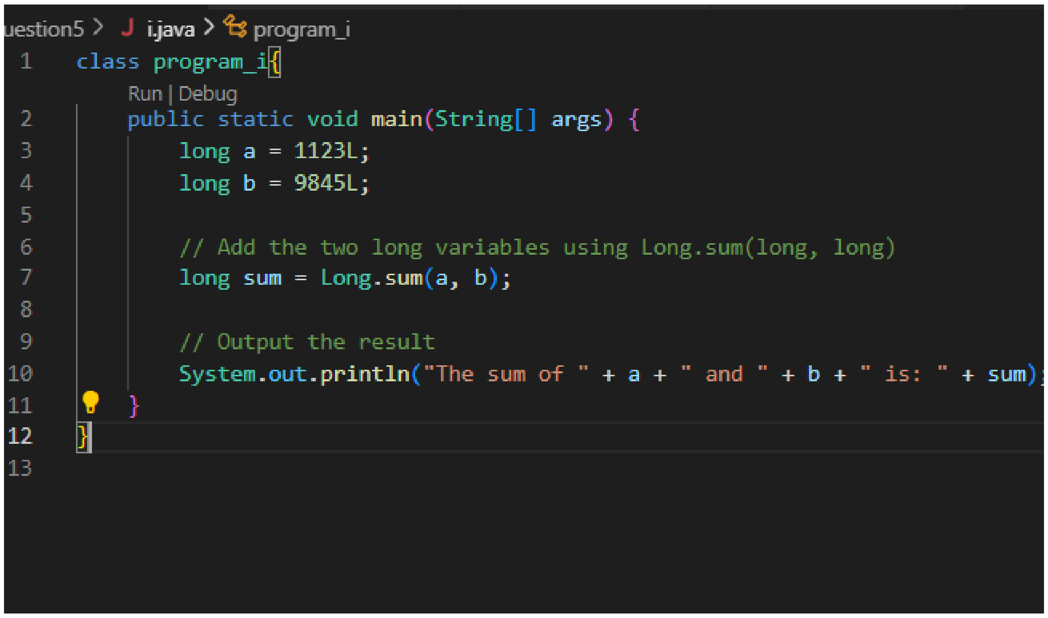
```
class program_h{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "123456789";

        // Convert the String to a Long object using Long.valueOf(String)
        try {
            Long longObject = Long.valueOf(strNumber);
            System.out.println("The Long object is: " + longObject);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid long value.");
        }
    }
}
```

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)`).

Ans. class program_i{

```
  public static void main(String[] args) {

    long a = 1123L;

    long b = 9845L;



    // Add the two long variables using Long.sum(long, long)

    long sum = Long.sum(a, b);



    // Output the result

    System.out.println("The sum of " + a + " and " + b + " is: " + sum);

  }
```

}



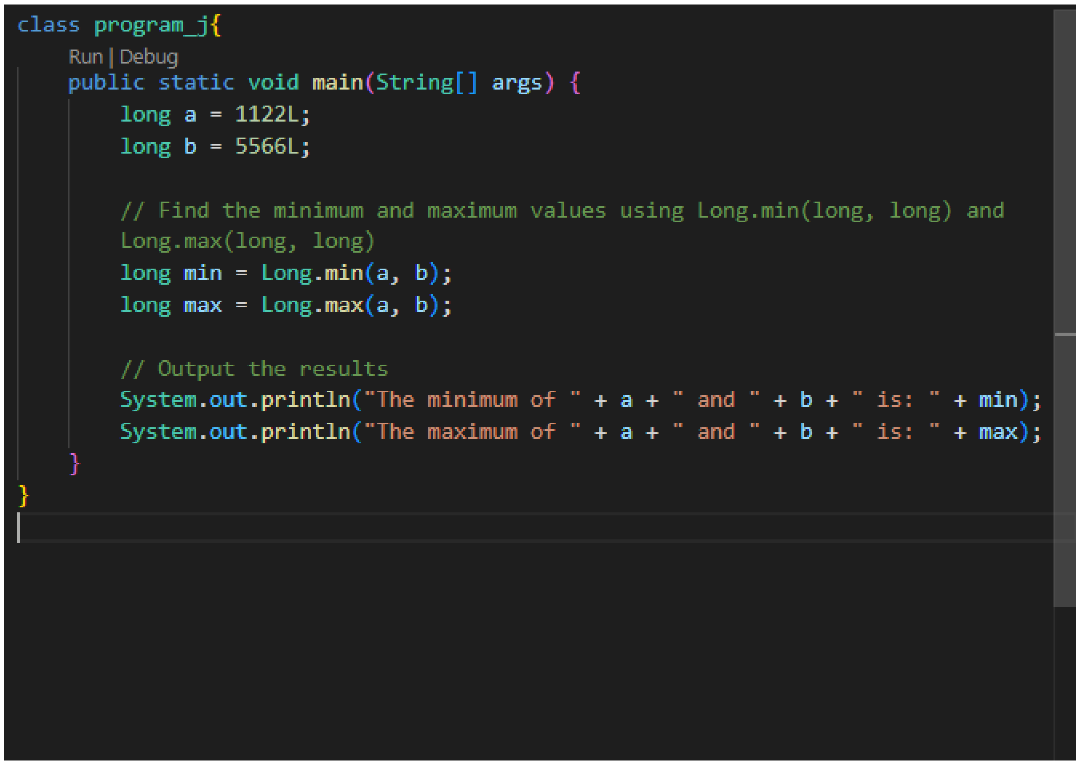
uestion5 > i.java > program_i

```
class program_i{
    Run | Debug
    public static void main(String[] args) {
        long a = 1123L;
        long b = 9845L;

        // Add the two long variables using Long.sum(long, long)
        long sum = Long.sum(a, b);

        // Output the result
        System.out.println("The sum of " + a + " and " + b + " is: " + sum)
    }
}
```

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)`).

Ans. class program_j{

```
    public static void main(String[] args) {

        long a = 1122L;

        long b = 5566L;



        // Find the minimum and maximum values using Long.min(long, long) and Long.max(long, long)

        long min = Long.min(a, b);

        long max = Long.max(a, b);



        // Output the results

        System.out.println("The minimum of " + a + " and " + b + " is: " + min);

        System.out.println("The maximum of " + a + " and " + b + " is: " + max);
    }

}
```

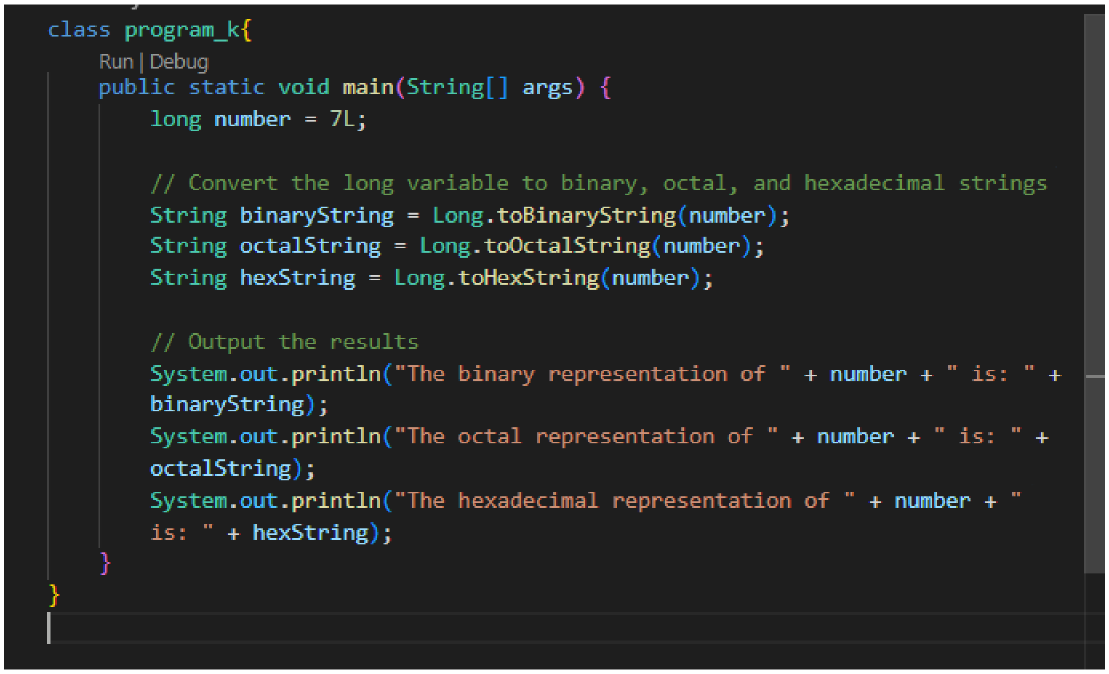


```
class program_j{
    Run | Debug
    public static void main(String[] args) {
        long a = 1122L;
        long b = 5566L;

        // Find the minimum and maximum values using Long.min(long, long) and
        Long.max(long, long)
        long min = Long.min(a, b);
        long max = Long.max(a, b);

        // Output the results
        System.out.println("The minimum of " + a + " and " + b + " is: " + min);
        System.out.println("The maximum of " + a + " and " + b + " is: " + max);
    }
}
```

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)`).

Ans. class program_k{

```
    public static void main(String[] args) {

        long number = 7L;



        // Convert the long variable to binary, octal, and hexadecimal strings

        String binaryString = Long.toBinaryString(number);

        String octalString = Long.toOctalString(number);
        String hexString = Long.toHexString(number);



        // Output the results

        System.out.println("The binary representation of " + number + " is: " + binaryString);

        System.out.println("The octal representation of " + number + " is: " + octalString);

        System.out.println("The hexadecimal representation of " + number + " is: " + hexString);

    }

}
```



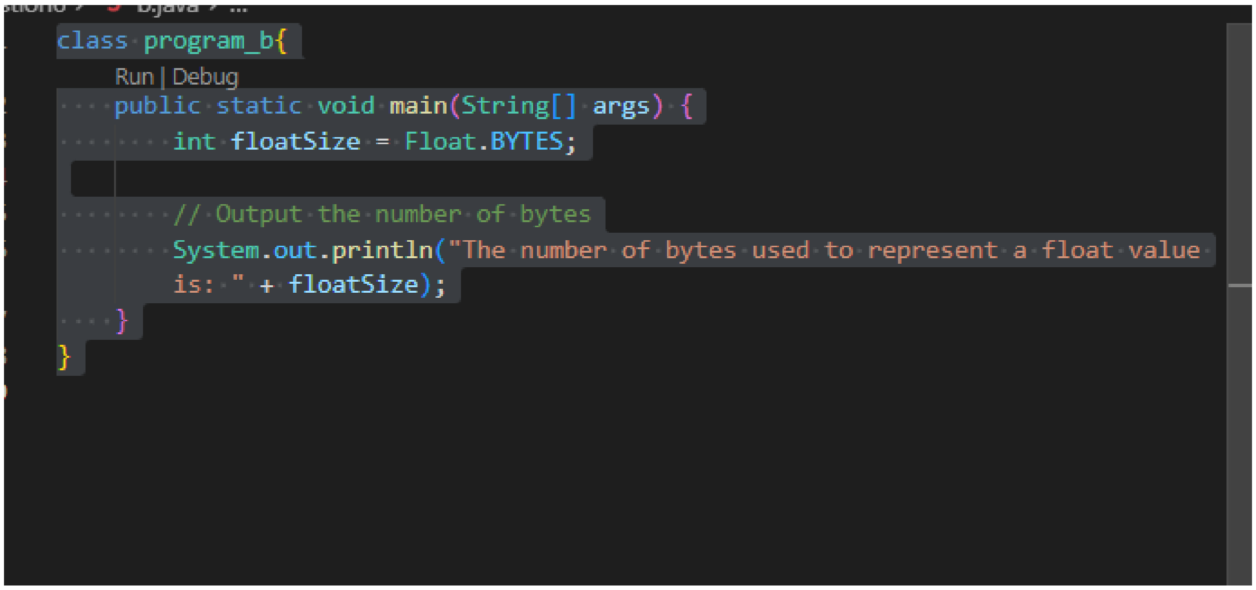
```
class program_k{
    Run | Debug
    public static void main(String[] args) {
        long number = 7L;

        // Convert the long variable to binary, octal, and hexadecimal strings
        String binaryString = Long.toBinaryString(number);
        String octalString = Long.toOctalString(number);
        String hexString = Long.toHexString(number);

        // Output the results
        System.out.println("The binary representation of " + number + " is: " +
        binaryString);
        System.out.println("The octal representation of " + number + " is: " +
        octalString);
        System.out.println("The hexadecimal representation of " + number + "
        is: " + hexString);
    }
}
```

6. Working with `java.lang.Float`

b. Write a program to test how many bytes are used to represent a `float` value using the `BYTES` field. (Hint: Use `Float.BYTES`).

Ans. class program_b{

```
  public static void main(String[] args) {

    int floatSize = Float.BYTES;



    // Output the number of bytes

    System.out.println("The number of bytes used to represent a float value is: " + floatSize);

  }

}
```



```
class program_b{
    Run | Debug
    public static void main(String[] args) {
        int floatSize = Float.BYTES;

        // Output the number of bytes
        System.out.println("The number of bytes used to represent a float value
        is: " + floatSize);
    }
}
```

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`).

Ans.

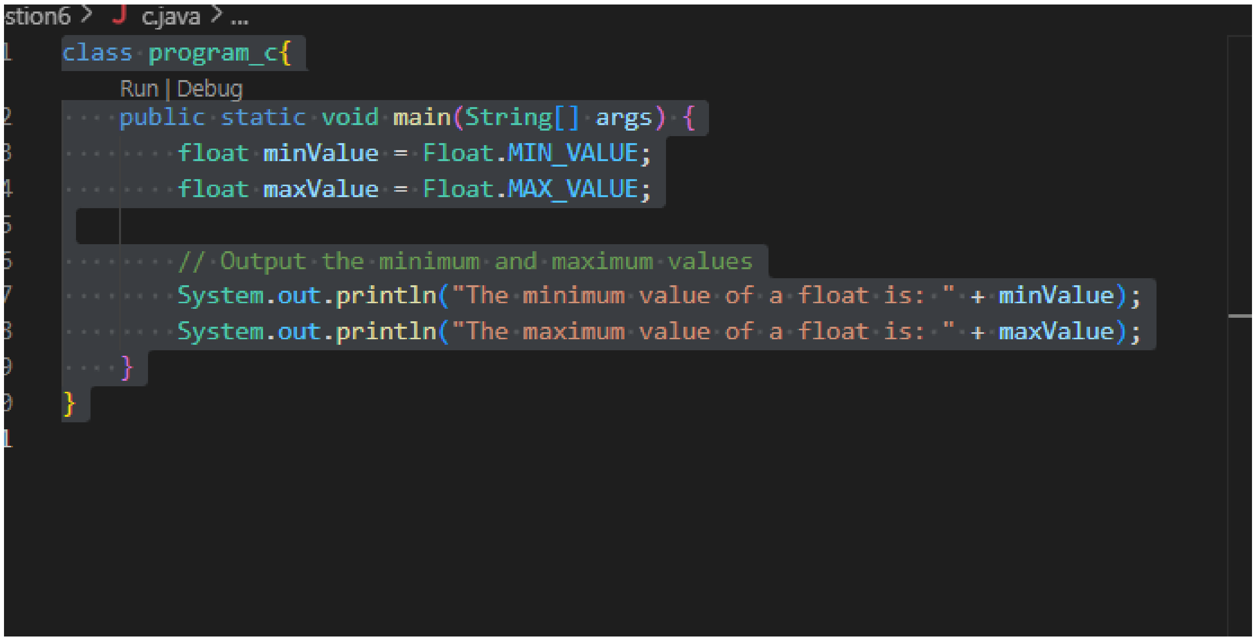
```
class program_c{

  public static void main(String[] args) {

    float minValue = Float.MIN_VALUE;

    float maxValue = Float.MAX_VALUE;



    // Output the minimum and maximum values
```

System.out.println("The minimum value of a float is: " + minValue);

System.out.println("The maximum value of a float is: " + maxValue);

}

}



stion6 > c.java > ...

```
class program_c{
    Run | Debug
    public static void main(String[] args) {
        float minValue = Float.MIN_VALUE;
        float maxValue = Float.MAX_VALUE;

        // Output the minimum and maximum values
        System.out.println("The minimum value of a float is: " + minValue);
        System.out.println("The maximum value of a float is: " + maxValue);
    }
}
```

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)`).

Ans. class program_d{

public static void main(String[] args) {

float number = 123.456f;

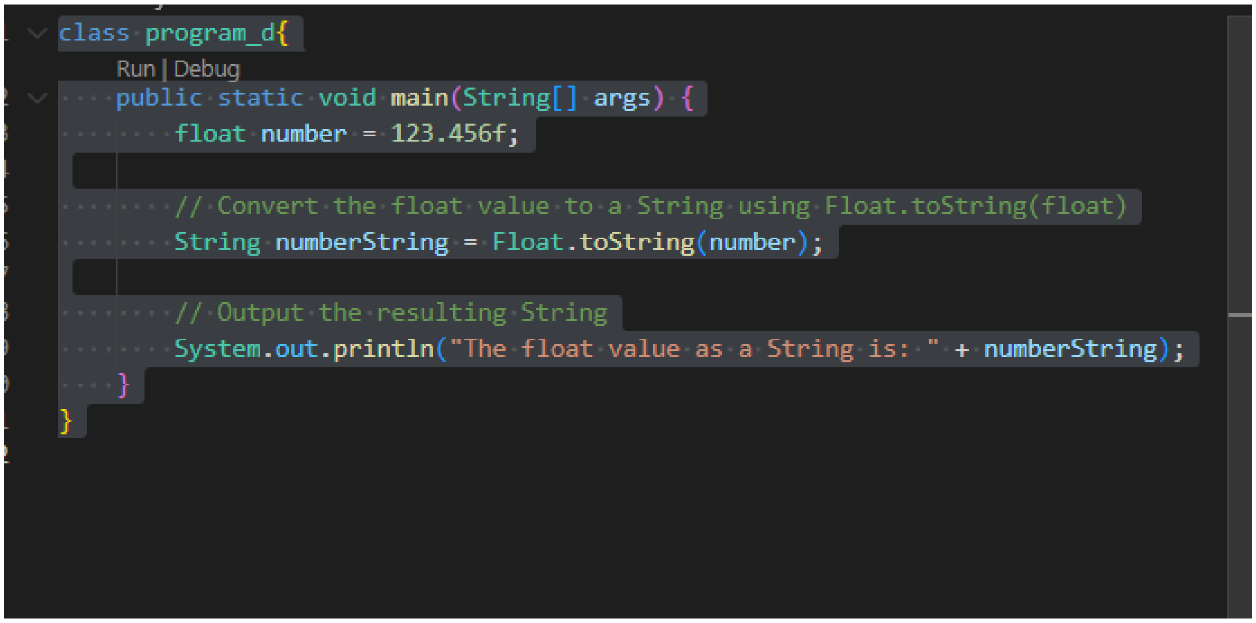
// Convert the float value to a String using Float.toString(float)

String numberString = Float.toString(number);

// Output the resulting String

System.out.println("The float value as a String is: " + numberString);

```
   }

}
```



```
class program_d{
    Run | Debug
    public static void main(String[] args) {
        float number = 123.456f;

        // Convert the float value to a String using Float.toString(float)
        String numberString = Float.toString(number);

        // Output the resulting String
        System.out.println("The float value as a String is: " + numberString);
    }
}
```

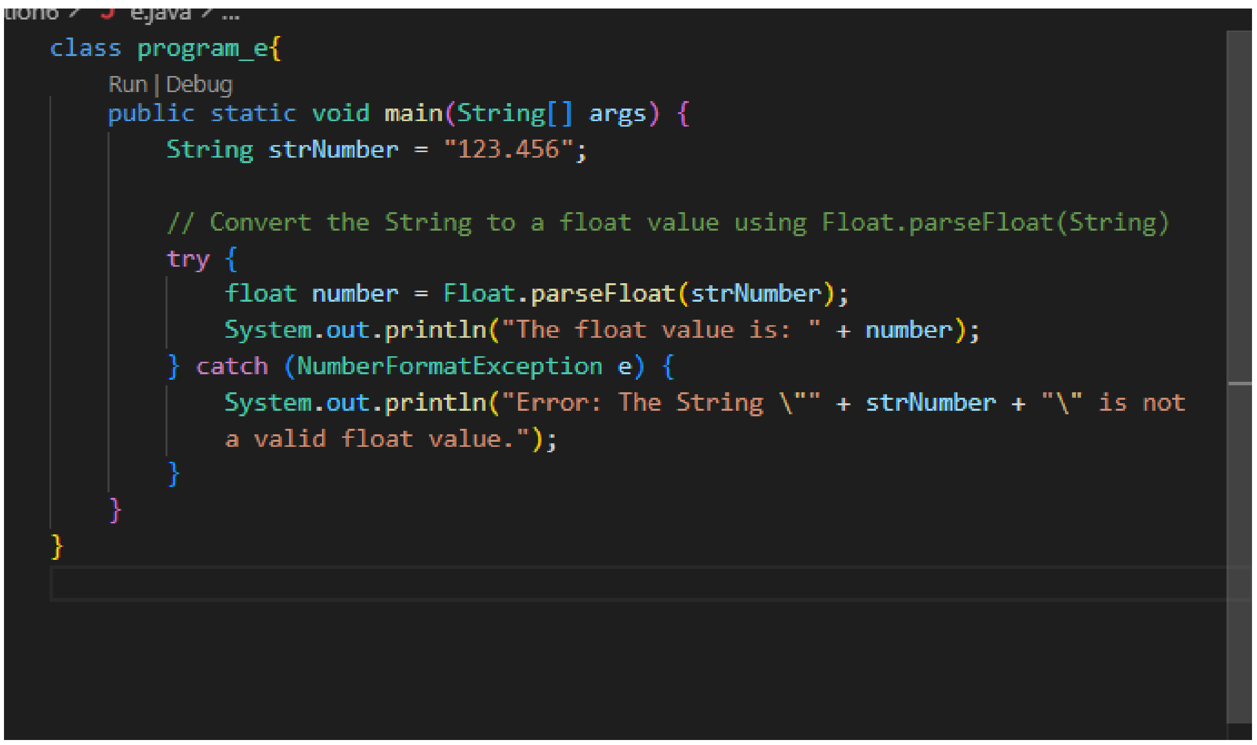
e. Declare a method-local variable `strNumber` of type `String` with some value and convert it to a `float` value using the `parseFloat` method. (Hint: Use `Float.parseFloat(String)`).

Ans.

```
class program_e{

   public static void main(String[] args) {

      String strNumber = "123.456";



      // Convert the String to a float value using Float.parseFloat(String)

      try {

         float number = Float.parseFloat(strNumber);

         System.out.println("The float value is: " + number);

      } catch (NumberFormatException e) {

         System.out.println("Error: The String \"" + strNumber + "\" is not a valid float value.");

      }
```

    }

}

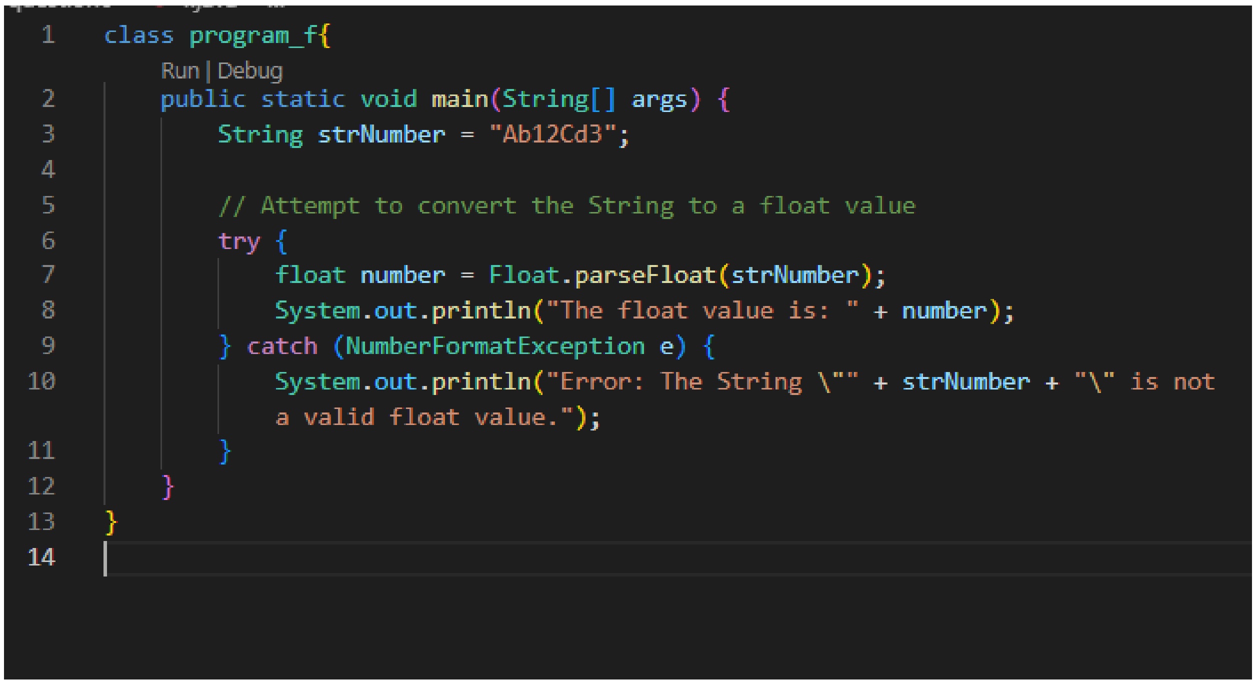


```
class program_e{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "123.456";

        // Convert the String to a float value using Float.parseFloat(String)
        try {
            float number = Float.parseFloat(strNumber);
            System.out.println("The float value is: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid float value.");
        }
    }
}
```

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`).

Ans. class program_f{

```
  public static void main(String[] args) {

    String strNumber = "Ab12Cd3";



    // Attempt to convert the String to a float value

    try {

      float number = Float.parseFloat(strNumber);

      System.out.println("The float value is: " + number);

    } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not a valid float value.");
        }
    }
}
```



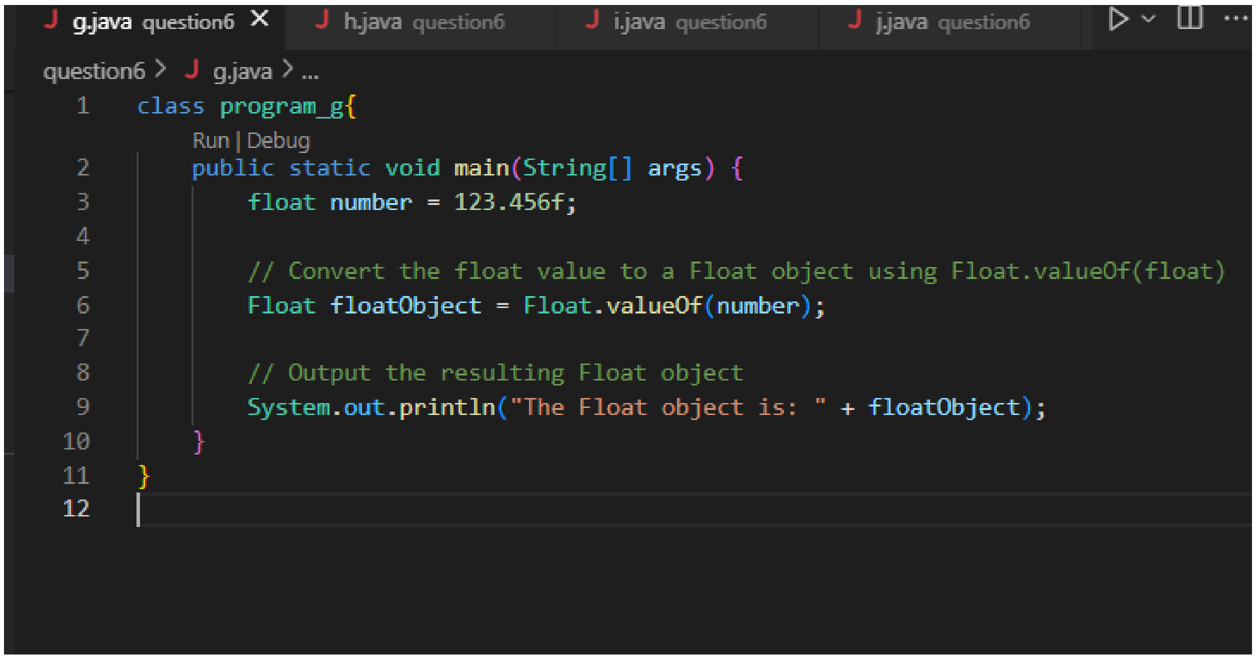
```
class program_f{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "Ab12Cd3";

        // Attempt to convert the String to a float value
        try {
            float number = Float.parseFloat(strNumber);
            System.out.println("The float value is: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid float value.");
        }
    }
}
```

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)`).

Ans. class program_g{

```
  public static void main(String[] args) {
    float number = 123.456f;


    // Convert the float value to a Float object using Float.valueOf(float)
    Float floatObject = Float.valueOf(number);


    // Output the resulting Float object
```

```
        System.out.println("The Float object is: " + floatObject);
    }
}
```



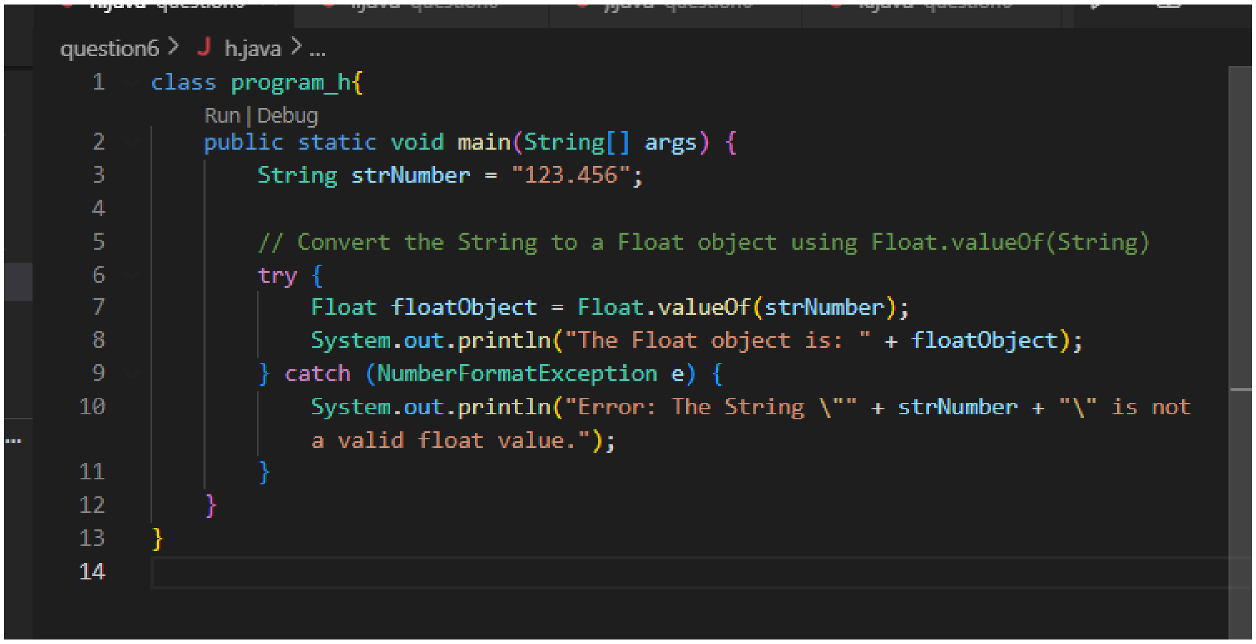
```
class program_g{
    public static void main(String[] args) {
        float number = 123.456f;

        // Convert the float value to a Float object using Float.valueOf(float)
        Float floatObject = Float.valueOf(number);

        // Output the resulting Float object
        System.out.println("The Float object is: " + floatObject);
    }
}
```

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)`).

Ans.

```
class program_h{
    public static void main(String[] args) {
        String strNumber = "123.456";


        // Convert the String to a Float object using Float.valueOf(String)
        try {
            Float floatObject = Float.valueOf(strNumber);
            System.out.println("The Float object is: " + floatObject);
        } catch (NumberFormatException e) {
```

```
        System.out.println("Error: The String \"" + strNumber + "\" is not a valid float value.");
    }
  }
}
```



question6 > h.java > ...

Run | Debug

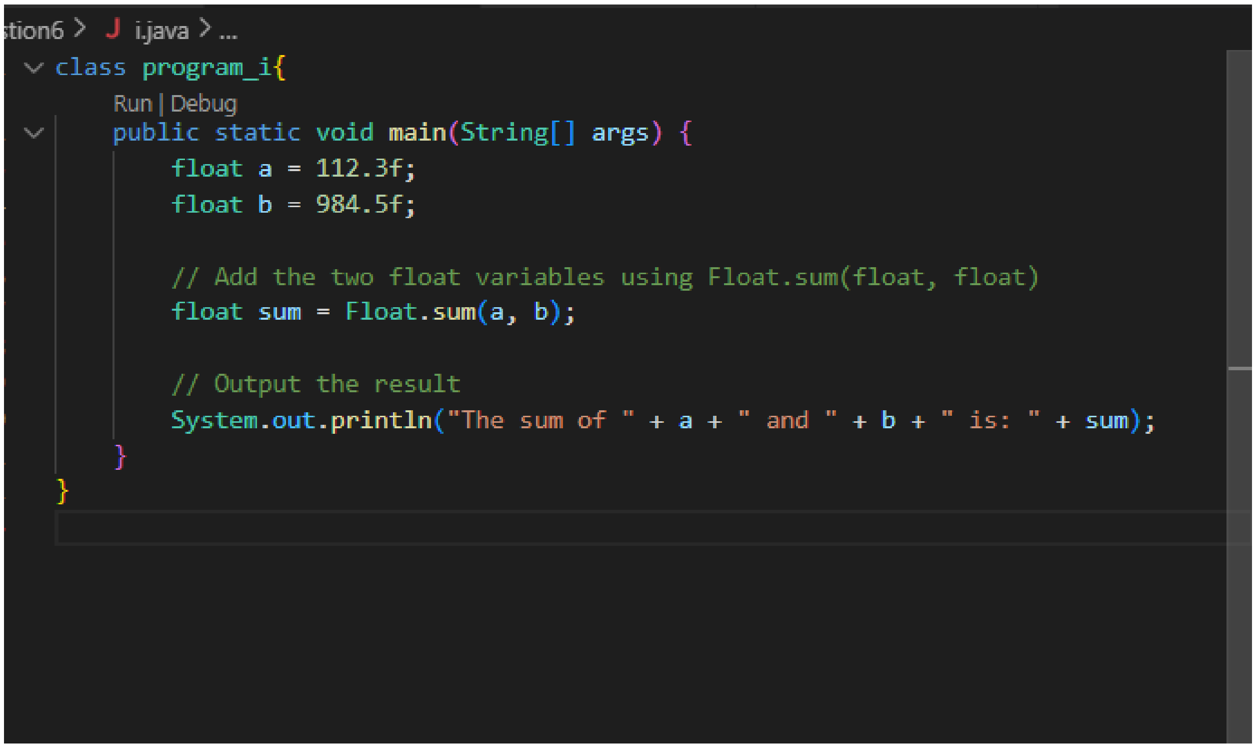
```
class program_h{
    public static void main(String[] args) {
        String strNumber = "123.456";

        // Convert the String to a Float object using Float.valueOf(String)
        try {
            Float floatObject = Float.valueOf(strNumber);
            System.out.println("The Float object is: " + floatObject);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid float value.");
        }
    }
}
```

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)`).

Ans. class program_i{

```
  public static void main(String[] args) {
    float a = 112.3f;
    float b = 984.5f;


    // Add the two float variables using Float.sum(float, float)
    float sum = Float.sum(a, b);


    // Output the result
```

```
        System.out.println("The sum of " + a + " and " + b + " is: " + sum);

    }

}
```



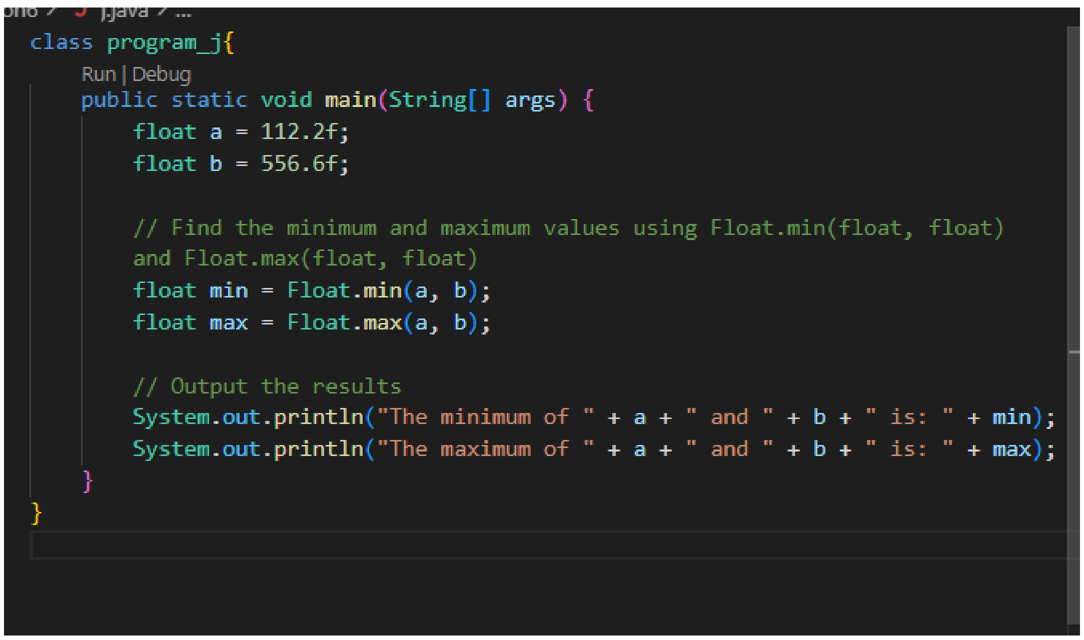
stion6 > J i.java > ...

```
class program_i{
    public static void main(String[] args) {
        float a = 112.3f;
        float b = 984.5f;

        // Add the two float variables using Float.sum(float, float)
        float sum = Float.sum(a, b);

        // Output the result
        System.out.println("The sum of " + a + " and " + b + " is: " + sum);
    }
}
```

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)`).

Ans. class program_j{

```
    public static void main(String[] args) {

        float a = 112.2f;

        float b = 556.6f;



        // Find the minimum and maximum values using Float.min(float, float) and Float.max(float, float)

        float min = Float.min(a, b);

        float max = Float.max(a, b);
        // Output the results

        System.out.println("The minimum of " + a + " and " + b + " is: " + min);

        System.out.println("The maximum of " + a + " and " + b + " is: " + max);

    }

}
```



```
class program_j{
    public static void main(String[] args) {
        float a = 112.2f;
        float b = 556.6f;

        // Find the minimum and maximum values using Float.min(float, float)
        and Float.max(float, float)
        float min = Float.min(a, b);
        float max = Float.max(a, b);

        // Output the results
        System.out.println("The minimum of " + a + " and " + b + " is: " + min);
        System.out.println("The maximum of " + a + " and " + b + " is: " + max);
    }
}
```

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

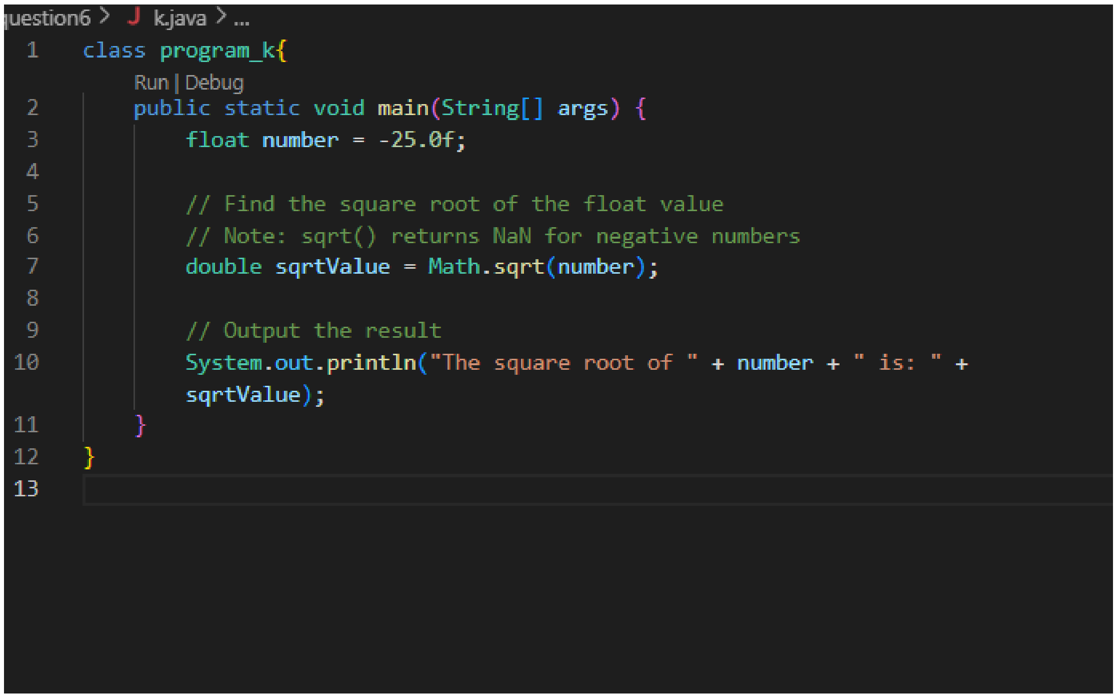
Ans.
```
class program_k{

    public static void main(String[] args) {

        float number = -25.0f;



        // Find the square root of the float value

        // Note: sqrt() returns NaN for negative numbers

        double sqrtValue = Math.sqrt(number);
```

        // Output the result

        System.out.println("The square root of " + number + " is: " + sqrtValue);

    }

}



uestion6 > k.java > ...

```java
class program_k{
    Run | Debug
    public static void main(String[] args) {
        float number = -25.0f;

        // Find the square root of the float value
        // Note: sqrt() returns NaN for negative numbers
        double sqrtValue = Math.sqrt(number);

        // Output the result
        System.out.println("The square root of " + number + " is: " +
        sqrtValue);
    }
}
```

7. Working with `java.lang.Double`

b. Write a program to test how many bytes are used to represent a `double` value using the `BYTES` field. (Hint: Use `Double.BYTES`).

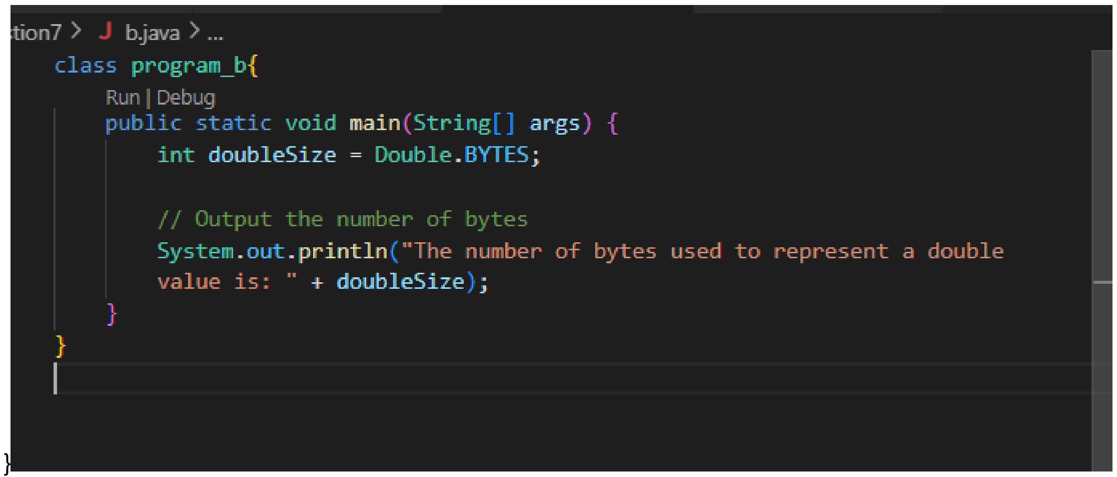
Ans.   class program_b{

public static void main(String[] args) {

    int doubleSize = Double.BYTES;

    // Output the number of bytes

```
    System.out.println("The number of bytes used to represent a double value is: " + doubleSize);

  }
```



tion7 > b.java > ...

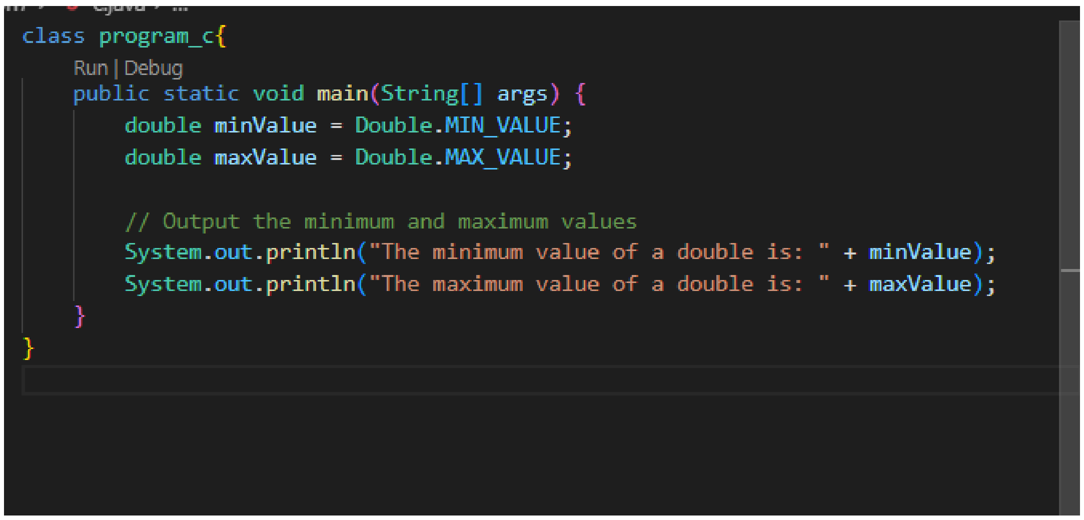
```
class program_b{
    Run | Debug
    public static void main(String[] args) {
        int doubleSize = Double.BYTES;

        // Output the number of bytes
        System.out.println("The number of bytes used to represent a double
        value is: " + doubleSize);
    }
}
```

}

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`).

Ans.

```
class program_c{

  public static void main(String[] args) {

    double minValue = Double.MIN_VALUE;

    double maxValue = Double.MAX_VALUE;



    // Output the minimum and maximum values

    System.out.println("The minimum value of a double is: " + minValue);

    System.out.println("The maximum value of a double is: " + maxValue);

  }

}
```

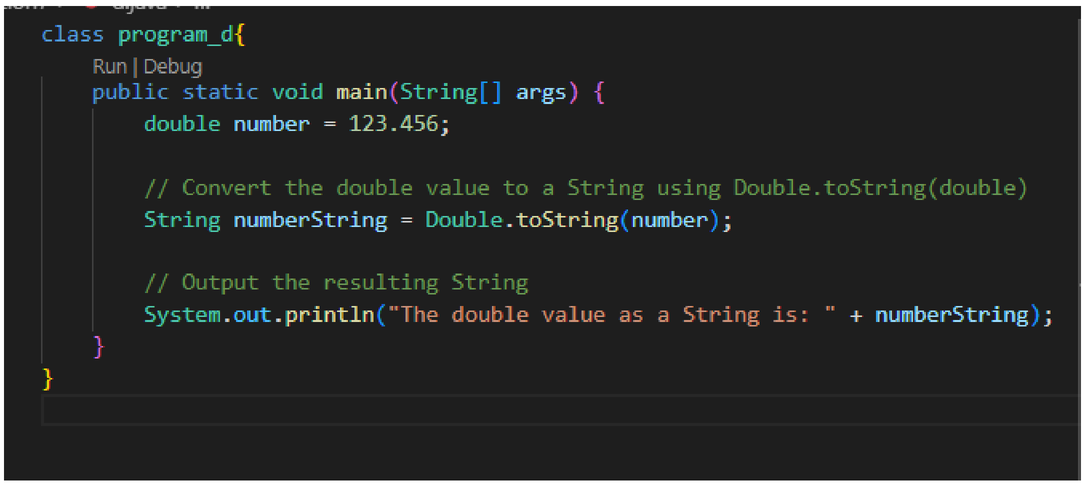


```
class program_c{
    Run | Debug
    public static void main(String[] args) {
        double minValue = Double.MIN_VALUE;
        double maxValue = Double.MAX_VALUE;

        // Output the minimum and maximum values
        System.out.println("The minimum value of a double is: " + minValue);
        System.out.println("The maximum value of a double is: " + maxValue);
    }
}
```

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)`).

Ans. class program_d{

```
  public static void main(String[] args) {

      double number = 123.456;



      // Convert the double value to a String using Double.toString(double)

      String numberString = Double.toString(number);



      // Output the resulting String

      System.out.println("The double value as a String is: " + numberString);

  }

}
```

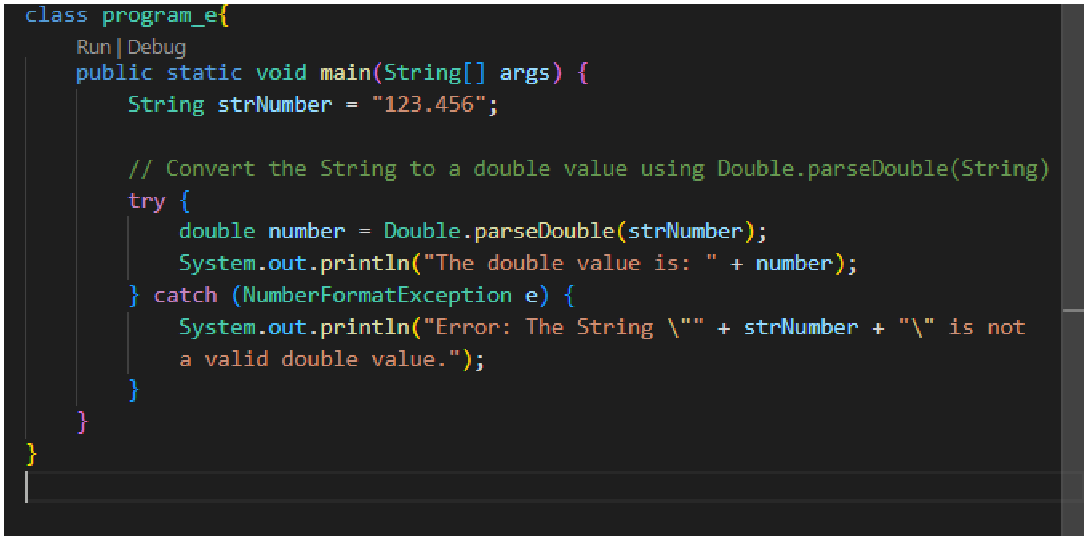


```
class program_d{
    Run | Debug
    public static void main(String[] args) {
        double number = 123.456;

        // Convert the double value to a String using Double.toString(double)
        String numberString = Double.toString(number);

        // Output the resulting String
        System.out.println("The double value as a String is: " + numberString);
    }
}
```

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)`).

Ans.

```
class program_e{

  public static void main(String[] args) {

    String strNumber = "123.456";



    // Convert the String to a double value using Double.parseDouble(String)

    try {

      double number = Double.parseDouble(strNumber);

      System.out.println("The double value is: " + number);

    } catch (NumberFormatException e) {

      System.out.println("Error: The String \"" + strNumber + "\" is not a valid double value.");

    }

  }

}
```



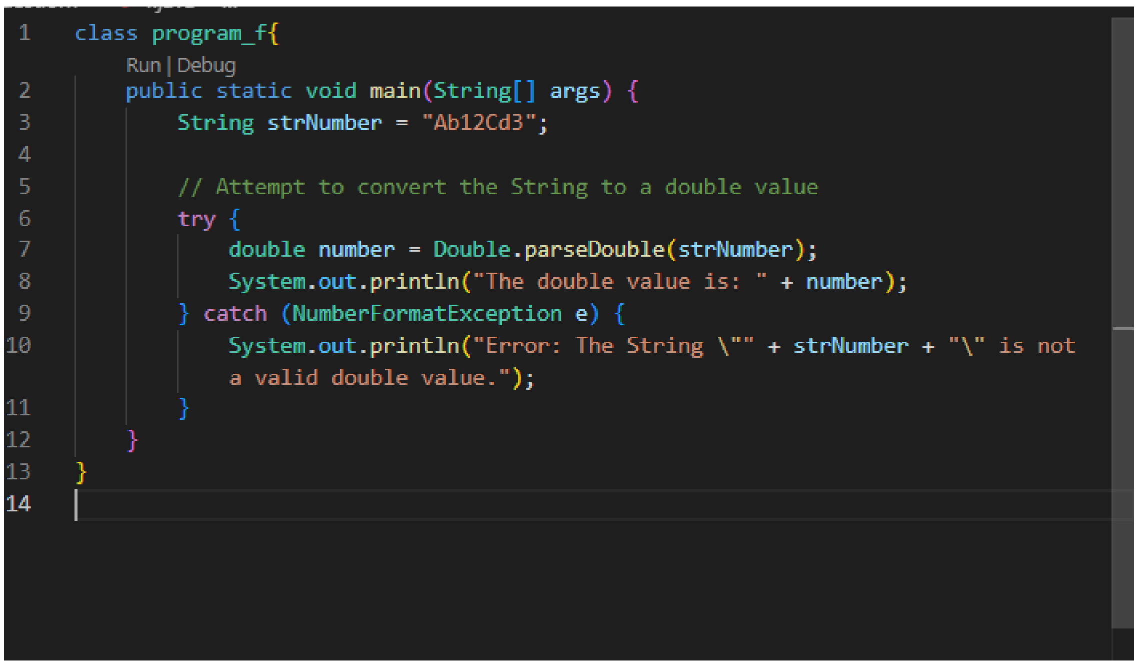
```
class program_e{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "123.456";

        // Convert the String to a double value using Double.parseDouble(String)
        try {
            double number = Double.parseDouble(strNumber);
            System.out.println("The double value is: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid double value.");
        }
    }
}
```

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`).

Ans.

```
class program_f{

  public static void main(String[] args) {

    String strNumber = "Ab12Cd3";



    // Attempt to convert the String to a double value

    try {

      double number = Double.parseDouble(strNumber);

      System.out.println("The double value is: " + number);

    } catch (NumberFormatException e) {

      System.out.println("Error: The String \"" + strNumber + "\" is not a valid double value.");

    }

  }

}
```

`

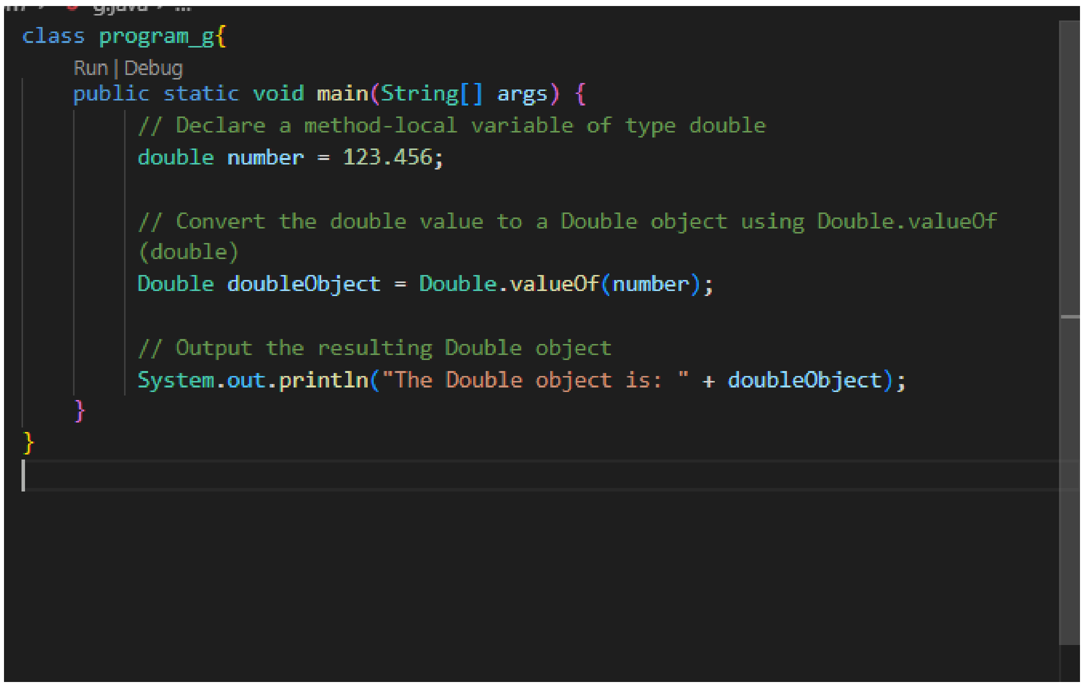


```java
class program_f{
    Run | Debug
    public static void main(String[] args) {
        String strNumber = "Ab12Cd3";

        // Attempt to convert the String to a double value
        try {
            double number = Double.parseDouble(strNumber);
            System.out.println("The double value is: " + number);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid double value.");
        }
    }
}
```

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)`).

Ans. class program_g{

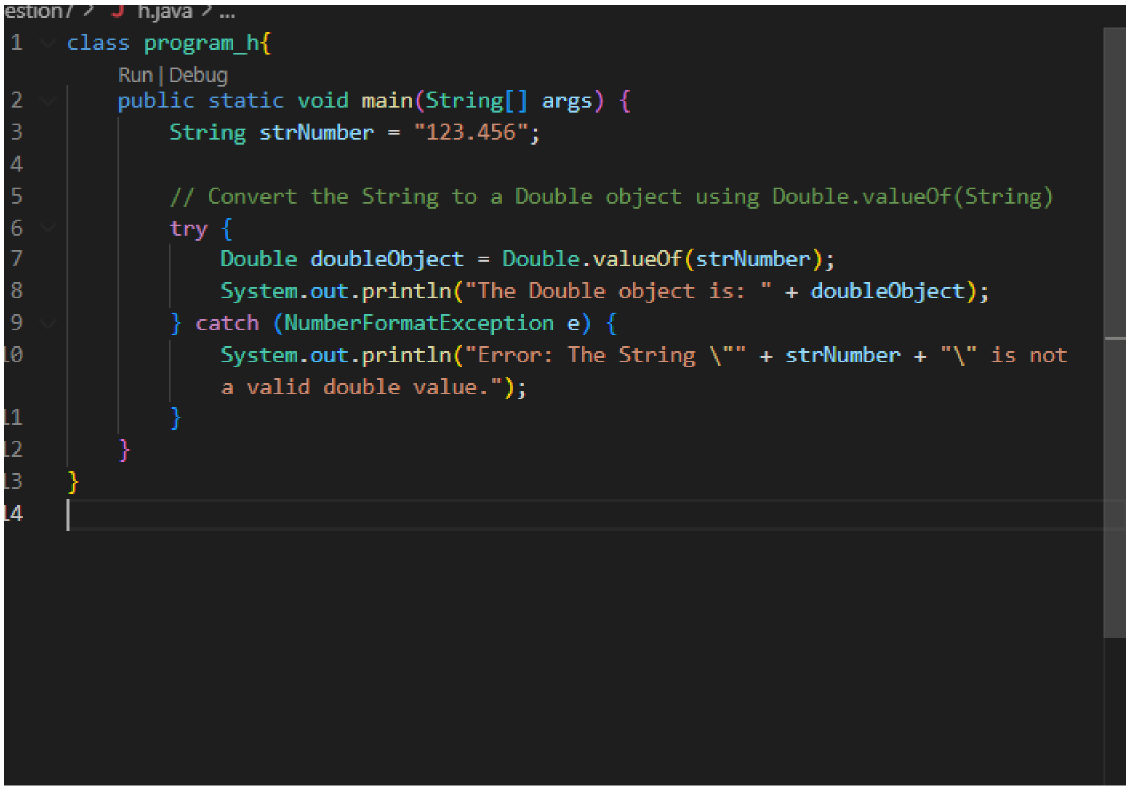
```
 public static void main(String[] args) {

     // Declare a method-local variable of type double

     double number = 123.456;



     // Convert the double value to a Double object using Double.valueOf(double)

     Double doubleObject = Double.valueOf(number);



     // Output the resulting Double object

     System.out.println("The Double object is: " + doubleObject);

 }

}
```



```
class program_g{
    Run | Debug
    public static void main(String[] args) {
        // Declare a method-local variable of type double
        double number = 123.456;

        // Convert the double value to a Double object using Double.valueOf
        (double)
        Double doubleObject = Double.valueOf(number);

        // Output the resulting Double object
        System.out.println("The Double object is: " + doubleObject);
    }
}
```

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)`).

Ans.
```
class program_h{
  public static void main(String[] args) {
    String strNumber = "123.456";


    // Convert the String to a Double object using Double.valueOf(String)
    try {
      Double doubleObject = Double.valueOf(strNumber);
      System.out.println("The Double object is: " + doubleObject);
    } catch (NumberFormatException e) {
      System.out.println("Error: The String \"" + strNumber + "\" is not a valid double value.");
    }
  }
}
```



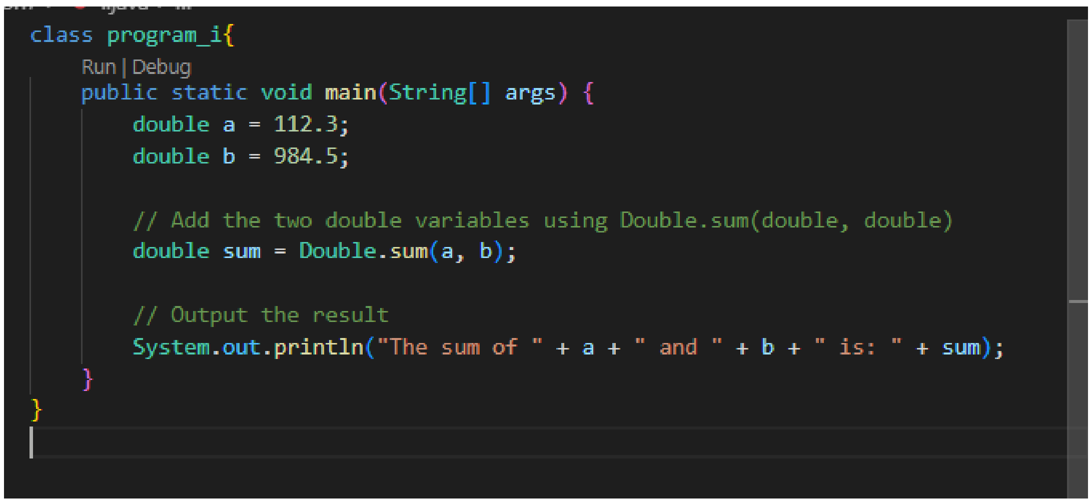
```java
class program_h{
    public static void main(String[] args) {
        String strNumber = "123.456";

        // Convert the String to a Double object using Double.valueOf(String)
        try {
            Double doubleObject = Double.valueOf(strNumber);
            System.out.println("The Double object is: " + doubleObject);
        } catch (NumberFormatException e) {
            System.out.println("Error: The String \"" + strNumber + "\" is not
            a valid double value.");
        }
    }
}
```

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)`).

Ans. class program_i{

```
  public static void main(String[] args) {

    double a = 112.3;

    double b = 984.5;



    // Add the two double variables using Double.sum(double, double)

    double sum = Double.sum(a, b);



    // Output the result

    System.out.println("The sum of " + a + " and " + b + " is: " + sum);

  }
```

}

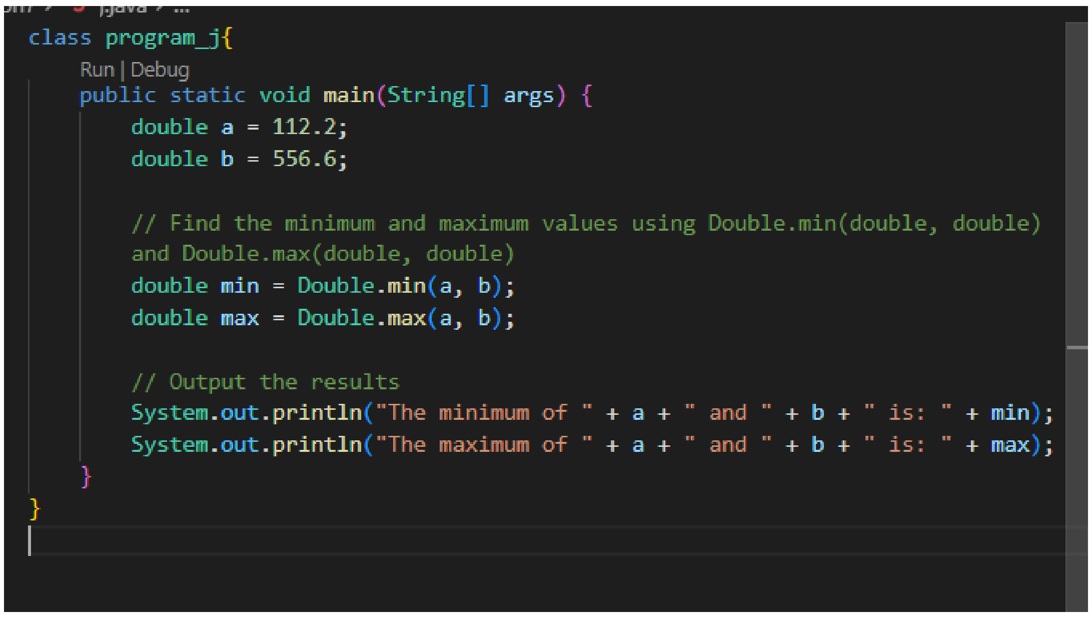


```
class program_i{
    Run | Debug
    public static void main(String[] args) {
        double a = 112.3;
        double b = 984.5;

        // Add the two double variables using Double.sum(double, double)
        double sum = Double.sum(a, b);

        // Output the result
        System.out.println("The sum of " + a + " and " + b + " is: " + sum);
    }
}
```

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)`).

Ans. class program_j{

```
  public static void main(String[] args) {

    double a = 112.2;

    double b = 556.6;



    // Find the minimum and maximum values using Double.min(double, double) and
Double.max(double, double)

    double min = Double.min(a, b);

    double max = Double.max(a, b);



    // Output the results

    System.out.println("The minimum of " + a + " and " + b + " is: " + min);

    System.out.println("The maximum of " + a + " and " + b + " is: " + max);

  }

}
```

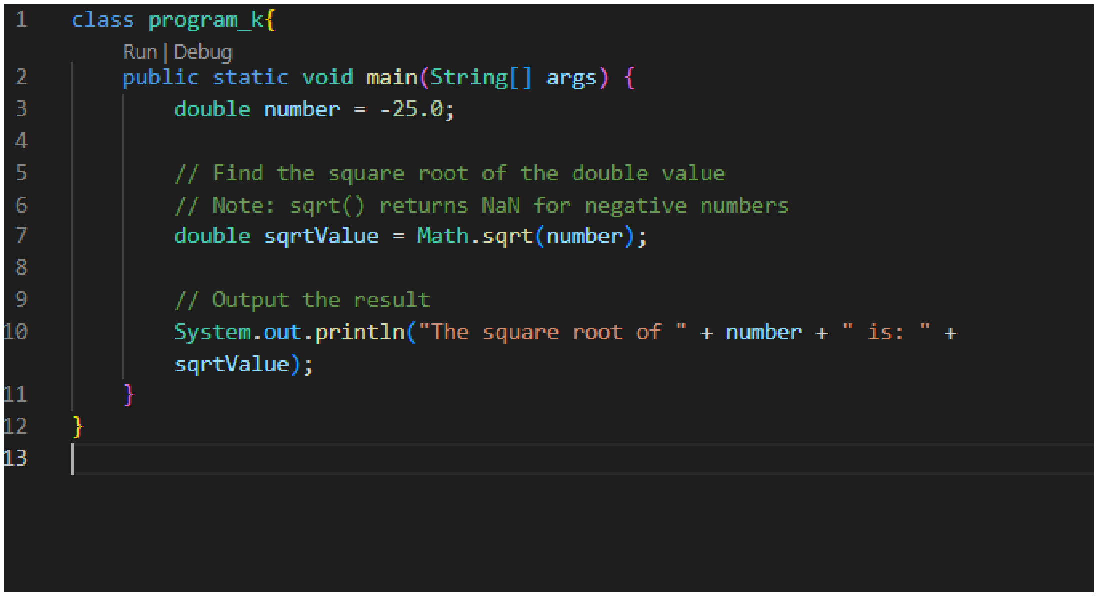


```
class program_j{
    Run | Debug
    public static void main(String[] args) {
        double a = 112.2;
        double b = 556.6;

        // Find the minimum and maximum values using Double.min(double, double)
        and Double.max(double, double)
        double min = Double.min(a, b);
        double max = Double.max(a, b);

        // Output the results
        System.out.println("The minimum of " + a + " and " + b + " is: " + min);
        System.out.println("The maximum of " + a + " and " + b + " is: " + max);
    }
}
```

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

Ans. class program_k{

```
  public static void main(String[] args) {

    double number = -25.0;



    // Find the square root of the double value

    // Note: sqrt() returns NaN for negative numbers

    double sqrtValue = Math.sqrt(number);



    // Output the result

    System.out.println("The square root of " + number + " is: " + sqrtValue);

  }

}
```


```
class program_k{
    public static void main(String[] args) {
        double number = -25.0;

        // Find the square root of the double value
        // Note: sqrt() returns NaN for negative numbers
        double sqrtValue = Math.sqrt(number);

        // Output the result
        System.out.println("The square root of " + number + " is: " +
        sqrtValue);
    }
}
```

8. Conversion between Primitive Types and Strings

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

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

Ans.

```
class PrimitiveToStringConversion {

    public static void main(String[] args) {

        // Initialize variables of each primitive type

        int intValue = 123;

        long longValue = 123456789L;

        float floatValue = 12.34f;

        double doubleValue = 123.456;

        char charValue = 'A';

        short shortValue = 12345;
```

```
byte byteValue = 123;

boolean booleanValue = true;


// Convert each primitive type to String using the toString method of the corresponding wrapper class

String intToString = Integer.toString(intValue);

String longToString = Long.toString(longValue);

String floatToString = Float.toString(floatValue);

String doubleToString = Double.toString(doubleValue);

String charToString = Character.toString(charValue);

String shortToString = Short.toString(shortValue);

String byteToString = Byte.toString(byteValue);

String booleanToString = Boolean.toString(booleanValue);


// Convert each primitive type to String using the valueOf method of the String class

String intValueOf = String.valueOf(intValue);

String longValueOf = String.valueOf(longValue);

String floatValueOf = String.valueOf(floatValue);

String doubleValueOf = String.valueOf(doubleValue);

String charValueOf = String.valueOf(charValue);

String shortValueOf = String.valueOf(shortValue);

String byteValueOf = String.valueOf(byteValue);

String booleanValueOf = String.valueOf(booleanValue);


// Output the results

System.out.println("Using toString method:");
```

```
        System.out.println("int to String: " + intToString);

        System.out.println("long to String: " + longToString);

        System.out.println("float to String: " + floatToString);

        System.out.println("double to String: " + doubleToString);

        System.out.println("char to String: " + charToString);

        System.out.println("short to String: " + shortToString);

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

        System.out.println("boolean to String: " + booleanToString);


        System.out.println("\nUsing valueOf method:");

        System.out.println("int to String: " + intValueOf);

        System.out.println("long to String: " + longValueOf);

        System.out.println("float to String: " + floatValueOf);

        System.out.println("double to String: " + doubleValueOf);

        System.out.println("char to String: " + charValueOf);

        System.out.println("short to String: " + shortValueOf);

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

        System.out.println("boolean to String: " + booleanValueOf);

    }

}
```

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).

```
Ans. class DefaultValuesExample {
    // Instance variables
    int instanceInt;
    long instanceLong;
    float instanceFloat;
    double instanceDouble;
    char instanceChar;
    short instanceShort;
    byte instanceByte;
    boolean instanceBoolean;

    // Static variables
    static int staticInt;
    static long staticLong;
    static float staticFloat;
    static double staticDouble;
    static char staticChar;
    static short staticShort;
    static byte staticByte;
    static boolean staticBoolean;

    public static void main(String[] args) {
        // Create an instance of the class to check instance variables
        DefaultValuesExample example = new DefaultValuesExample();
        // Output default values of instance variables
        System.out.println("Default values of instance variables:");
        System.out.println("int: " + example.instanceInt);
        System.out.println("long: " + example.instanceLong);
        System.out.println("float: " + example.instanceFloat);
        System.out.println("double: " + example.instanceDouble);
        System.out.println("char: [" + example.instanceChar + "]"); // char defaults to '\u0000', which is an
empty character
        System.out.println("short: " + example.instanceShort);
        System.out.println("byte: " + example.instanceByte);
        System.out.println("boolean: " + example.instanceBoolean);


        // Output default values of static variables
        System.out.println("\nDefault values of static variables:");
        System.out.println("int: " + DefaultValuesExample.staticInt);
        System.out.println("long: " + DefaultValuesExample.staticLong);
        System.out.println("float: " + DefaultValuesExample.staticFloat);
        System.out.println("double: " + DefaultValuesExample.staticDouble);
        System.out.println("char: [" + DefaultValuesExample.staticChar + "]"); // char defaults to '\u0000',
which is an empty character
        System.out.println("short: " + DefaultValuesExample.staticShort);
        System.out.println("byte: " + DefaultValuesExample.staticByte);
        System.out.println("boolean: " + DefaultValuesExample.staticBoolean);
    }
}
```

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).

Ans.
```
class ArithmeticOperations {
  public static void main(String[] args) {
    // Check if the correct number of arguments are provided
    if (args.length != 3) {
      System.out.println("Usage: java ArithmeticOperations <number1> <number2> <operator>");
      System.out.println("Example: java ArithmeticOperations 10 20 +");
      return;
    }

    // Parse command line arguments
    int number1;
    int number2;
    String operator = args[2];

    try {
      number1 = Integer.parseInt(args[0]);
      number2 = Integer.parseInt(args[1]);
    } catch (NumberFormatException e) {
      System.out.println("Error: The first two arguments must be integers.");
      return;
    }

    // Perform arithmetic operation based on the operator
    double result;
    switch (operator) {
      case "+":
        result = number1 + number2;
        break;
      case "-":
        result = number1 - number2;
        break;
      case "*":
        result = number1 * number2;
        break;
      case "/":
        if (number2 == 0) {
          System.out.println("Error: Division by zero is not allowed.");
          return;
        }
        result = (double) number1 / number2;
        break;
      default:
        System.out.println("Error: Unsupported operator. Please use +, -, *, or /.");
        return;
        }

        // Output the result
        System.out.printf("Result: %d %s %d = %.2f%n", number1, operator, number2, result);
    }
}
```