Lab 1

Faculty of Engineering and Applied Science

SOFE 3980U : Software Quality

Due: January 29th, 2024

Name: Daniel Amasowomwan
ID: 100787640
Email: daniel.amasowomwan@ontariotechu.net

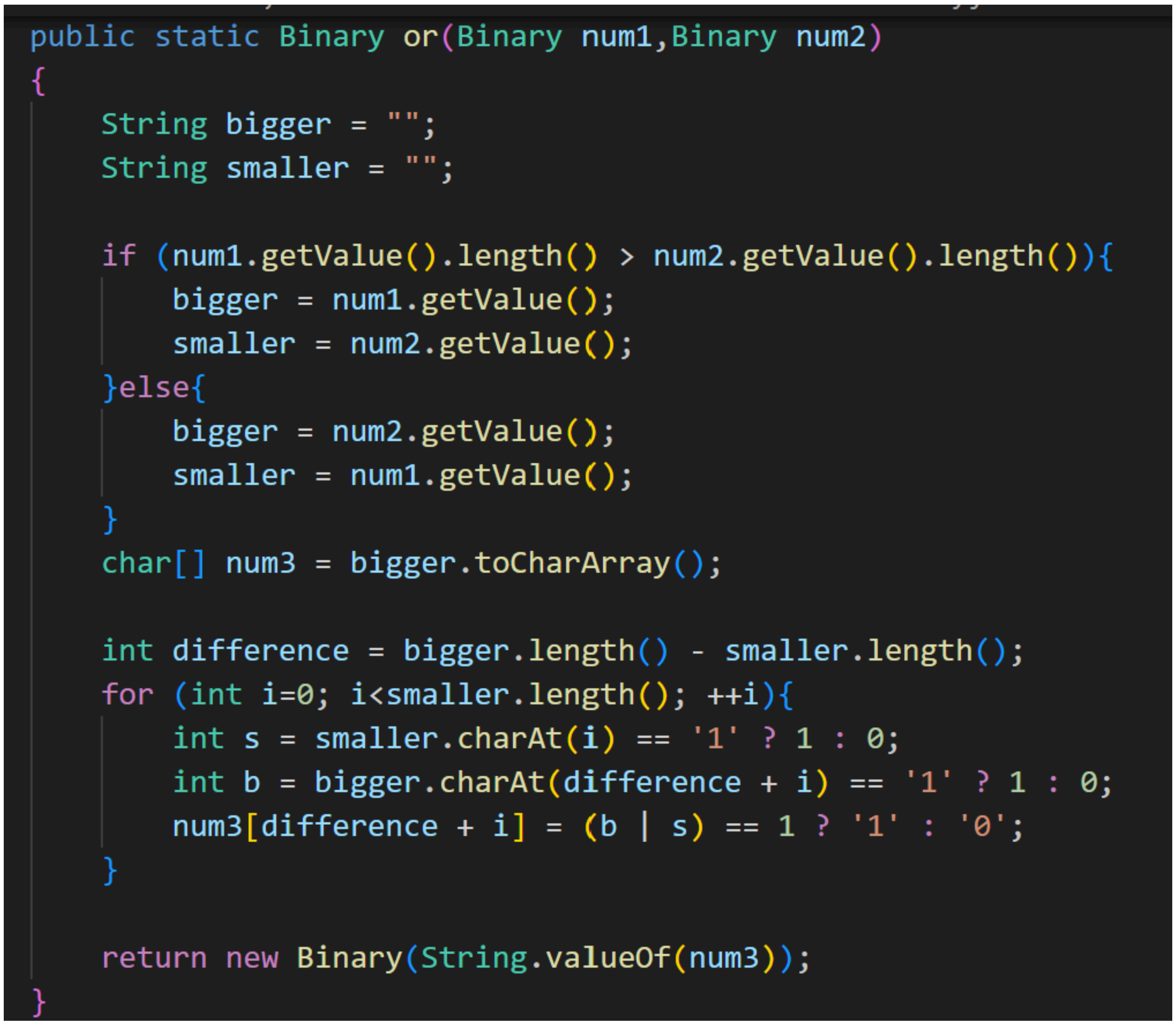
Github Repository Link: https://github.com/Daniel-Amas/SOFE3980U-Quality--Lab1.git

Video Link:
https://drive.google.com/file/d/1hX7yh1VDZ1xlhRMd6QTzDR7rHVUebXgG/view?usp=drive_link

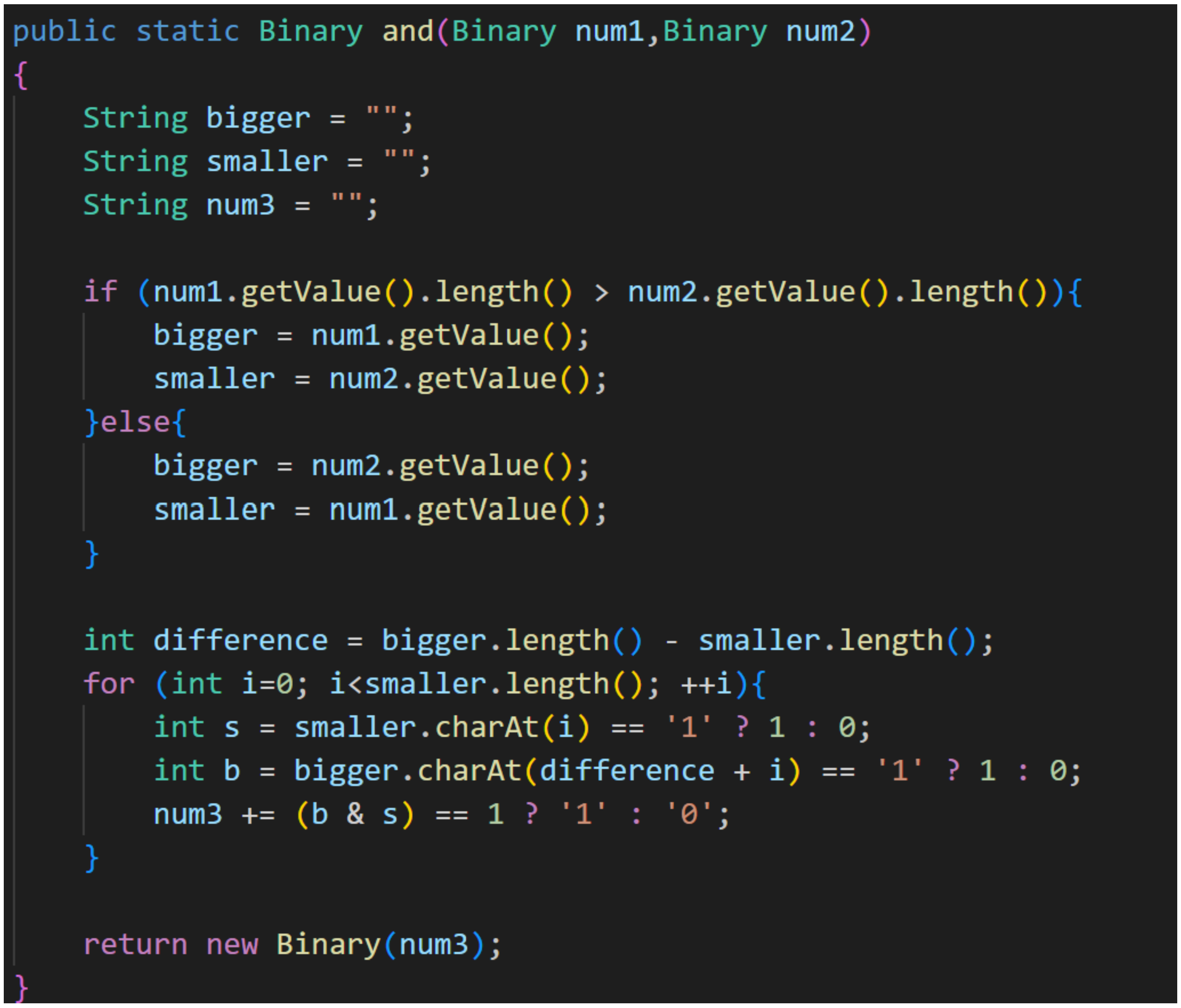
Introduction

The goal of this activity is to get familiar with Maven as a tool for managing software projects. It involves learning to create, configure, and build Maven projects, generating project documentation, setting up project dependencies, as well as writing and executing tests within the project.

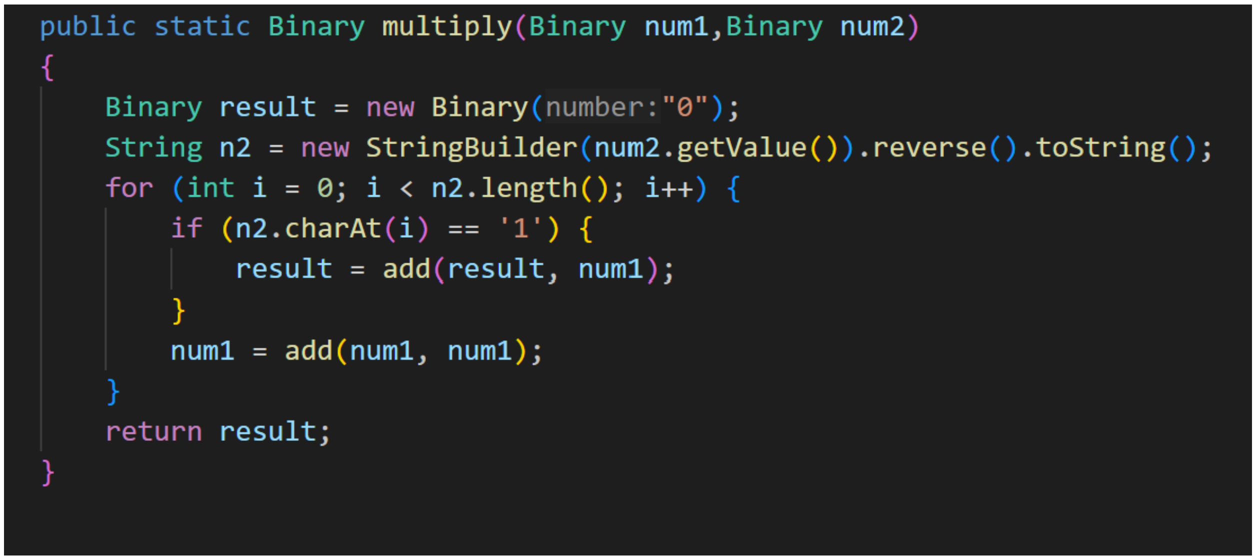
Binary Classes



```java
public static Binary or(Binary num1,Binary num2)
{
    String bigger = "";
    String smaller = "";

    if (num1.getValue().length() > num2.getValue().length()){
        bigger = num1.getValue();
        smaller = num2.getValue();
    }else{
        bigger = num2.getValue();
        smaller = num1.getValue();
    }
    char[] num3 = bigger.toCharArray();

    int difference = bigger.length() - smaller.length();
    for (int i=0; i<smaller.length(); ++i){
        int s = smaller.charAt(i) == '1' ? 1 : 0;
        int b = bigger.charAt(difference + i) == '1' ? 1 : 0;
        num3[difference + i] = (b | s) == 1 ? '1' : '0';
    }

    return new Binary(String.valueOf(num3));
}
```

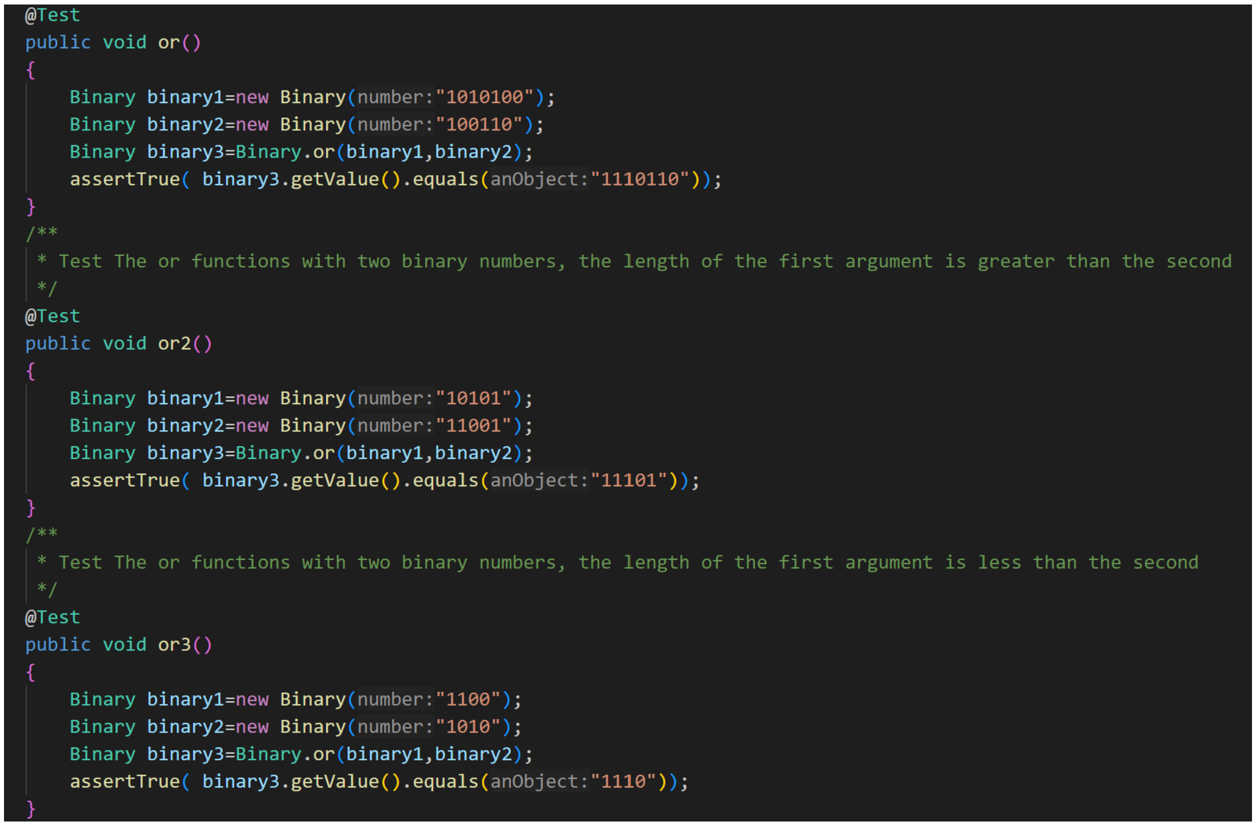


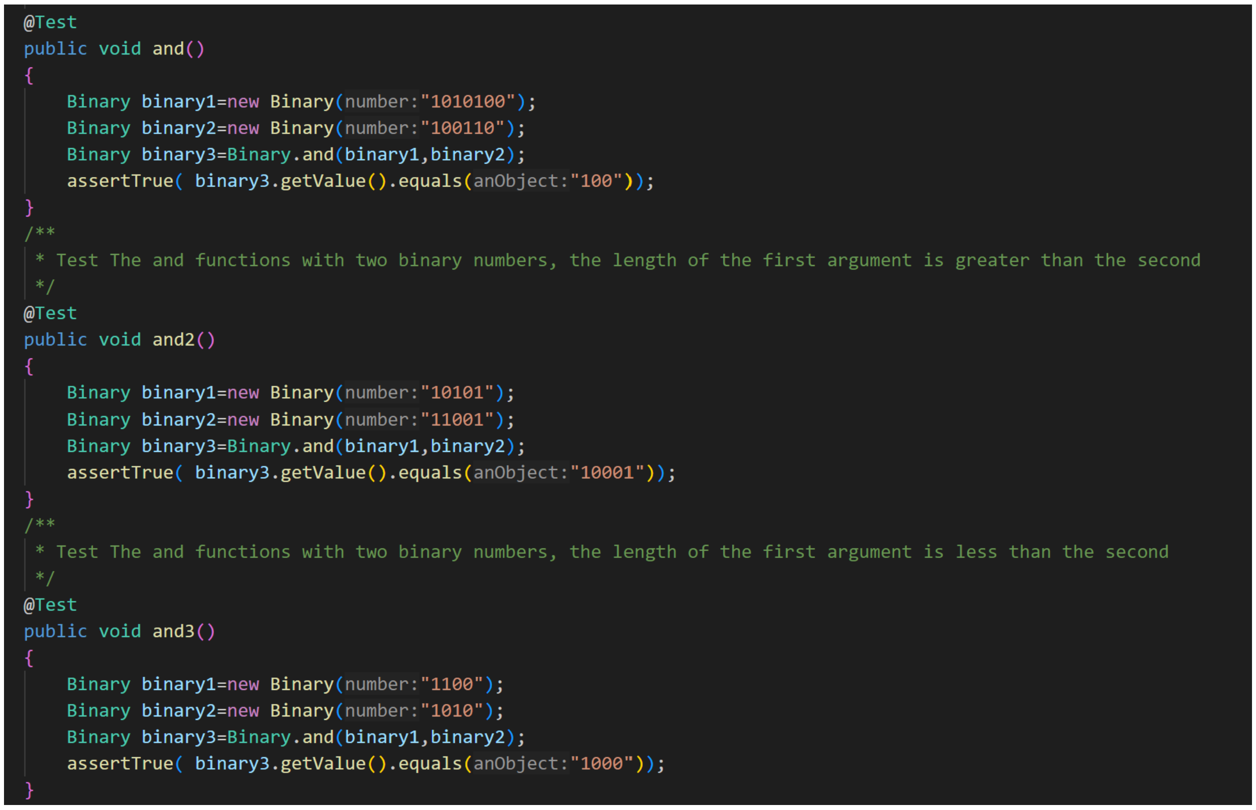
```java
public static Binary and(Binary num1,Binary num2)
{
    String bigger = "";
    String smaller = "";
    String num3 = "";

    if (num1.getValue().length() > num2.getValue().length()){
        bigger = num1.getValue();
        smaller = num2.getValue();
    }else{
        bigger = num2.getValue();
        smaller = num1.getValue();
    }

    int difference = bigger.length() - smaller.length();
    for (int i=0; i<smaller.length(); ++i){
        int s = smaller.charAt(i) == '1' ? 1 : 0;
        int b = bigger.charAt(difference + i) == '1' ? 1 : 0;
        num3 += (b & s) == 1 ? '1' : '0';
    }

    return new Binary(num3);
}
```

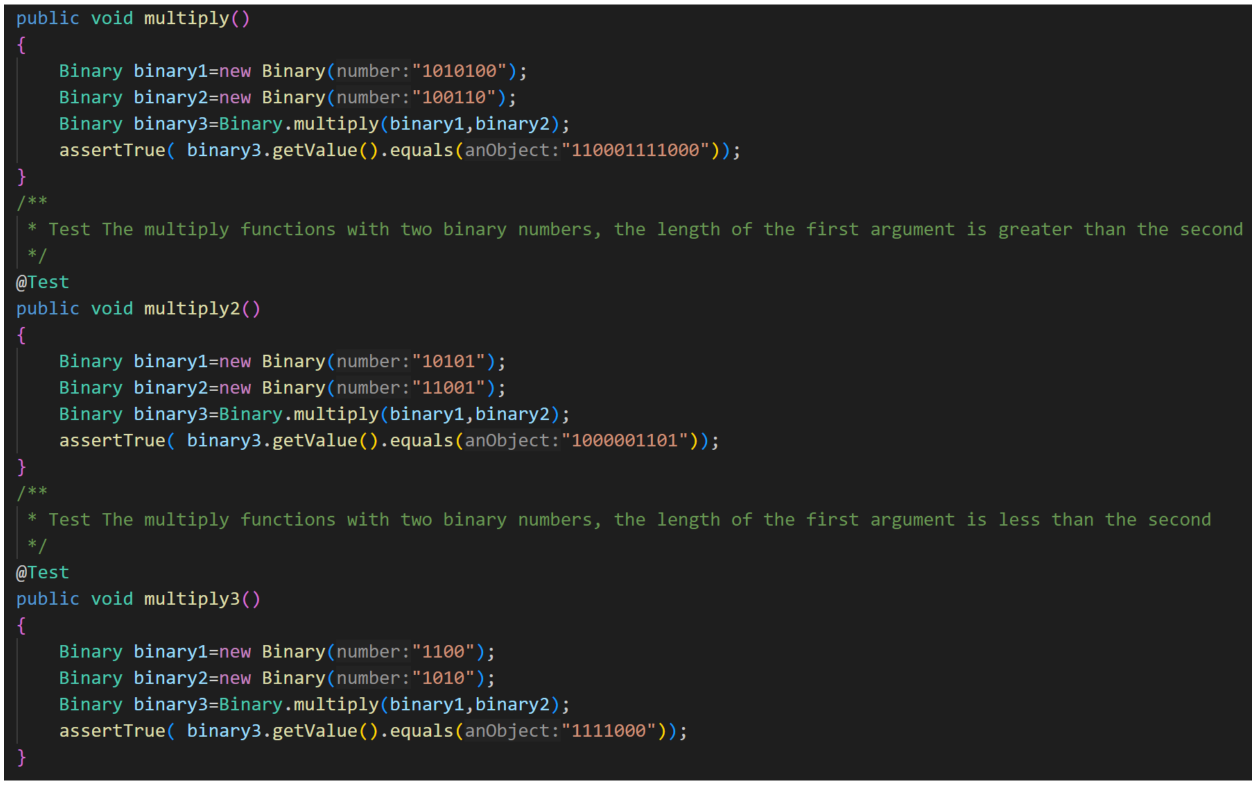


```java
public static Binary multiply(Binary num1,Binary num2)
{
    Binary result = new Binary(number:"0");
    String n2 = new StringBuilder(num2.getValue()).reverse().toString();
    for (int i = 0; i < n2.length(); i++) {
        if (n2.charAt(i) == '1') {
            result = add(result, num1);
        }
        num1 = add(num1, num1);
    }
    return result;
}
```

Test Cases

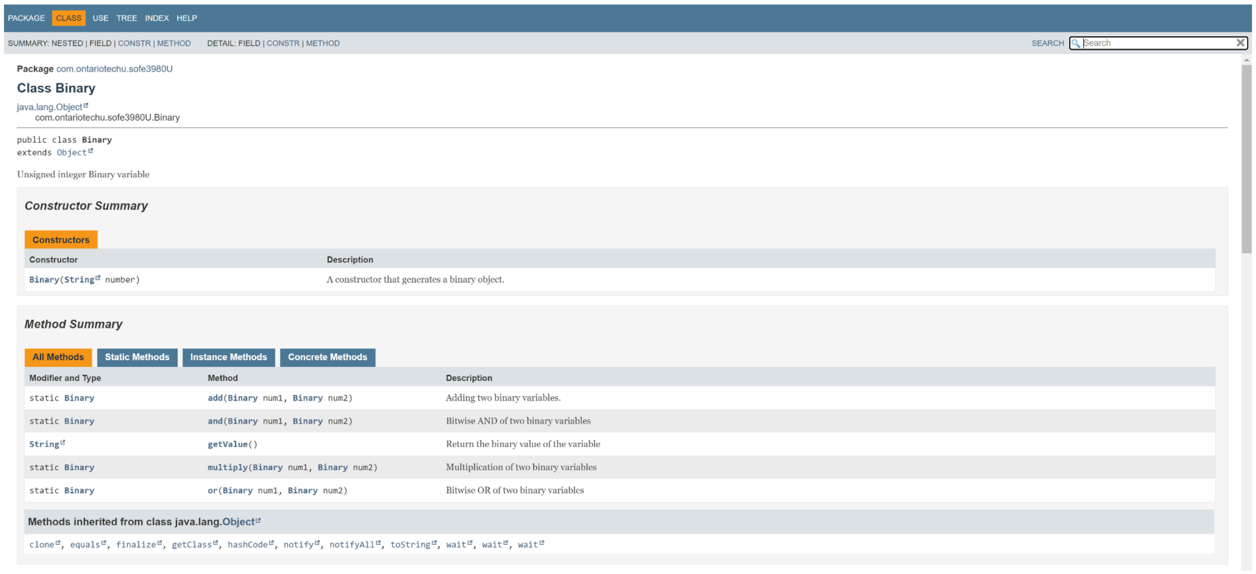

```
@Test
public void or()
{
    Binary binary1=new Binary(number:"1010100");
    Binary binary2=new Binary(number:"100110");
    Binary binary3=Binary.or(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"1110110"));
}
/**
 * Test The or functions with two binary numbers, the length of the first argument is greater than the second
 */
@Test
public void or2()
{
    Binary binary1=new Binary(number:"10101");
    Binary binary2=new Binary(number:"11001");
    Binary binary3=Binary.or(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"11101"));
}
/**
 * Test The or functions with two binary numbers, the length of the first argument is less than the second
 */
@Test
public void or3()
{
    Binary binary1=new Binary(number:"1100");
    Binary binary2=new Binary(number:"1010");
    Binary binary3=Binary.or(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"1110"));
}
```


```
@Test
public void and()
{
    Binary binary1=new Binary(number:"1010100");
    Binary binary2=new Binary(number:"100110");
    Binary binary3=Binary.and(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"100"));
}
/**
 * Test The and functions with two binary numbers, the length of the first argument is greater than the second
 */
@Test
public void and2()
{
    Binary binary1=new Binary(number:"10101");
    Binary binary2=new Binary(number:"11001");
    Binary binary3=Binary.and(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"10001"));
}
/**
 * Test The and functions with two binary numbers, the length of the first argument is less than the second
 */
@Test
public void and3()
{
    Binary binary1=new Binary(number:"1100");
    Binary binary2=new Binary(number:"1010");
    Binary binary3=Binary.and(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"1000"));
}
```



```java
public void multiply()
{
    Binary binary1=new Binary(number:"1010100");
    Binary binary2=new Binary(number:"100110");
    Binary binary3=Binary.multiply(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"110001111000"));
}
/**
 * Test The multiply functions with two binary numbers, the length of the first argument is greater than the second
 */
@Test
public void multiply2()
{
    Binary binary1=new Binary(number:"10101");
    Binary binary2=new Binary(number:"11001");
    Binary binary3=Binary.multiply(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"1000001101"));
}
/**
 * Test The multiply functions with two binary numbers, the length of the first argument is less than the second
 */
@Test
public void multiply3()
{
    Binary binary1=new Binary(number:"1100");
    Binary binary2=new Binary(number:"1010");
    Binary binary3=Binary.multiply(binary1,binary2);
    assertTrue( binary3.getValue().equals(anObject:"1111000"));
}
```

JavaDoc


PACKAGE CLASS USE TREE INDEX HELP
SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD
SEARCH
Search

Package com.ontariotechu.sofe3980U

Class Binary

java.lang.Object
com.ontariotechu.sofe3980U.Binary

```
public class Binary
extends Object
```

Unsigned integer Binary variable

Constructor Summary

Constructors

Constructor	Description
Binary(String number)	A constructor that generates a binary object.

Method Summary

All Methods | Static Methods | Instance Methods | Concrete Methods

Modifier and Type	Method	Description
static Binary	add(Binary num1, Binary num2)	Adding two binary variables.
static Binary	and(Binary num1, Binary num2)	Bitwise AND of two binary variables
String	getValue()	Return the binary value of the variable
static Binary	multiply(Binary num1, Binary num2)	Multiplication of two binary variables
static Binary	or(Binary num1, Binary num2)	Bitwise OR of two binary variables

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Surefire Report

Test Cases

[Summary] [Package List] [Test Cases]

AppTest

✓	shouldAnswerWithTrue	0

BinaryTest

✓	constructorWithInvalidDigits	0
✓	multiply2	0
✓	multiply3	0
✓	or	0
✓	add	0
✓	and	0
✓	or2	0
✓	or3	0
✓	add2	0
✓	add3	0
✓	add4	0
✓	add5	0
✓	and2	0
✓	and3	0
✓	constructorWithNegativeSign	0
✓	constructorWithInvalidChars	0
✓	constructorEmptyString	0
✓	multiply	0
✓	normalConstructor	0
✓	constructorWithZeroTailing	0.016