



# CSA 1017

## Data Structures and Algorithms 1

### Assignment

Mark Said Camilleri  
B.Sc. (Hons) (Computing Science)  
University of Malta

Friday 4th May 2016

# Contents

	Page
Task 1	3
Task 2	7
References	7

# Statement of Completion

The questions below were the ones that have been attempted:

**Question 1** This question has been successfully completed.

**Question 2** This question has been successfully completed.

**Question 3** This question has been successfully completed.

**Question 4** This question has been successfully completed.

**Question 5** This question has been successfully completed.

**Question 6** This question has been successfully completed.

**Question 7** This question has been successfully completed.

**Question 8** This question has been successfully completed.

**Question 9** This question has been successfully completed.

---

Signature

---

Date



# Task 1

Input	Expected Output	Actual Output
1	I	I
2	II	II
4	IV	IV
5	V	V
6	VI	IV
9	IX	IX
10	X	X
20	XX	XX
40	XL	XL
49	XLIX	XLIX
50	L	L
60	LX	LX
90	XC	XC
99	XCIX	XCIX
100	C	C
150	CL	CL
400	CD	CD
499	CDXCIX	CDXCIX
500	D	D
600	DC	DC
900	CM	CM
999	CMXCIX	CMXCIX
1000	M	M
1024	MXXIV	MXXIV
2000	The input was not a valid number between 1 and 1024 Please try again and enter a number to convert:	The input was not a valid number between 1 and 1024 Please try again and enter a number to convert:
abc	The input was not a valid number between 1 and 1024 Please try again and enter a number to convert:	The input was not a valid number between 1 and 1024 Please try again and enter a number to convert:

## The source code for Task 1

```
1  /**
2   * @author Mark Said Camilleri
3   * @version 20160509
4   */
5
6
7  import java.util.InputMismatchException;
8  import java.util.Scanner;
9
10
11 public class Question1 {
12
13     public static void main(String args[]) {
14
15         //Initialize Scanner object
16         Scanner in = new Scanner(System.in);
17         in.useDelimiter("\n");
18
19         /*===== WELCOME MESSAGE TO USER
20         =====*/
21         System.out.println("
22         -----");
23         System.out.println("|      CSA 1017 - Data Structures and
24         Algorithms 1      |");
25         System.out.println("
26         |-----|");
27         System.out.println("|      Submission by Mark Said Camilleri
28         |");
29         System.out.println("|      Task 1: Arabic to Roman Numeral
30         Converter      |");
31         System.out.println("
32         |-----|");
33         System.out.print("| Please enter a number between 1 and 1024: ")
34         ;
35
36
37         int toConvert = 0; //value to be converted.
38         boolean isError; // temporary boolean value used for error
39         checking of the input.
40         do {
41             isError = false;
42             try {
43                 toConvert = in.nextInt();
44             } catch (InputMismatchException e) {
45                 isError = true;
46                 in.next(); //To clear the buffer
47             }
48             /*===== Makes sure input is a number is between 1 and
49             1024 =====*/
50             if (isError || toConvert < 1 || toConvert > 1024) {
51                 /*===== OUTPUT ERROR MESSAGE TO THE USER
52                 =====*/
53                 System.out.println("
54                 |-----|");
55                 System.out.println("|The input was not a valid number
56                 between 1 and 1024|");
57             }
58         } while (isError || toConvert < 1 || toConvert > 1024);
59     }
60 }
```

```
44         System.out.print("|Please try again and enter a number
to convert: ");
45     }
46     } while (isError || toConvert < 1 || toConvert > 1024);
47
48     System.out.printf("| %4d = %-24s in Roman Numerals |", toConvert
, convert(toConvert));
49
50 }
51
52 /**
53  * Takes an int decimal value and outputs a string of the same value
in Roman Numerals.
54  *
55  * @param toConvert the decimal value to be converted to Roman
Numerals
56  * @return The roman numeral equivalent of the input parameter
57  */
58 private static String convert(int toConvert) {
59
60     //Defining the decimal and roman counterparts
61     final int dec[] = {1, 4, 5, 9, 10, 40, 50, 90, 100, 400, 500,
900, 1000};
62     final String rom[] = {"I", "IV", "V", "IX", "X", "XL", "L", "XC"
, "C", "CD", "D", "CM", "M"};
63
64     /* Begins by checking the input parameter against the largest
roman numeral/numeral pair.
65     * and works it's way down to the unit numeral.
66     */
67     for (int i = dec.length - 1; i >= 0; i--) {
68         //If the value is larger, then the output is concatenated
with the output of the difference.
69         if (toConvert >= dec[i])
70             return rom[i] + convert(toConvert - dec[i]);
71     }
72     return ""; //What to return at 0, the base case.
73 }
74 }
```

## Task 2

Assumption Made: *This program assumes that the expression provided at is a correct expression.*

Input	Expected Output	Actual Output
4 5 +	9.0	9.0
12 6 -	6.0	6.0
3 2 /	1.5	1.5
7 2 *	14.0	14.0
23 85 + 92 *	9936.0	9936.0
43.5 3.2 - 4.5 * 3.24 +	184.59	184.59
34.8 62.11 * -76 /	-28.43984211	-28.439842105263157
282 -56 * 1024 - -55.6 *	934969.6	934969.6
Test	Your expression contained invalid characters. For input string "T" Your expression is invalid. Evaluation failed	Your expression contained invalid characters. For input string "T" Your expression is invalid. Evaluation failed
1 +	Stack is Empty. Your expression is invalid. Evaluation failed.	Stack is Empty. Your expression is invalid. Evaluation failed.
3 64 6 +	The stack has not been emptied. There are too many operands in your expression. Your expression is invalid. Evaluation failed.	The stack has not been emptied. There are too many operands in your expression. Your expression is invalid. Evaluation failed.
4 0 /	Infinity	Infinity