

Computer Programming & Maths for Computing and IT

Module Code:

Module Title:	Computer Programming & Maths for Computing and IT		
Assignment Type:	Practical – Individual Project		
Project Title:	Binary Number Calculator		
Project Launch Date:	30/11/2017		
Due Date:	02/02/2018 at 23.59 – submission link via Moodle ONLY		
Assignment Compiler(s):	Carole McGloughlin, carole@cct.ie Kyle Goslin, kyle@cct.ie		
Weighting:	Maths for Computing and IT:	22%	
	Computer Programming:	15%	
Method of Submission:	Moodle online Submission [<i>Submission link Computer Programming Moodle page</i>]		
Feedback Method:	Maths for Computing & IT: rubric Computer Programming: rubric and comments PDF feedback file.		
Module Learning Outcome:	Maths for Computing and IT: <ul style="list-style-type: none">• Associate the mathematical rules to areas of mathematics and computing;• Approach and solve computing problems using the correct mathematical methods; Computer Programming: <ul style="list-style-type: none">• Implement the core syntax and semantics of the programming language utilised• Identify and use good principles of algorithm design to write and debug structured programs• Build robust software that adheres to current conventions, that is reliable, maintainable, available and if necessary expandable and reusable		

Assignment Introduction

This assignment makes up a portion of your assessment for both Computer Programming and Maths for Computing and IT. Each module's assessment components are clearly outlined in this document and while related, they are not double graded.

You are required to develop a binary number calculator using your binary number knowledge and your Java programming skills. Your code should be developed **yourself** and not use in-built binary library methods.

The program should allow a user to:

- Enter a decimal or binary number (positive or negative, real or integer) and provide the following arithmetic options: addition, subtraction, multiplication or division.
- Conversion to / from binary / decimal should also be provided and real numbers dealt with properly.
- The option to check your answer your decimal arithmetic should also be provided.
- The program should be developed from scratch, that is, do not use in built binary number libraries.
- Decimal numbers should be stored appropriate format.
- Binary numbers should be stored as a String, using 8 or 16 bits as appropriate, allowing each digit to be accessed and dealt separately.
- The option to use signed or unsigned binary representation should be provided, but where necessary signed representation (for all numbers involved) must be used. That is a two's complement conversion for negative numbers

Specific Requirements

Your program should be modular and contain methods to allow the following:

- A simple welcome message that presents the user with the following options in a Menu System:
 - A) Binary to decimal
 - B) Decimal to binary
 - C) Addition
 - D) Subtraction
 - E) Multiplication
 - F) Division
 - G) Quit
- After the menu has been show, the user should then be able to type in A,B,C,D,E,F or G only. If the user enters anything else, an error message should be shown and the menu items should be shown again.
- Option A should allow the user to enter a binary number that will then be converted to a decimal number.
- Option B should allow the user to enter in a decimal number that will then be converted to a binary number.
- After the user has entered a number in binary or decimal format, an option should exist to allow the user to store the converted value as either VARIABLE_1 or VARIABLE_2 which will be used later during the math operations.
- Calculations should be performed on the two stored input variables using dedicated custom methods for addition, subtraction, subtraction and division. Each method should calculate the answer and print it to the console.
- Individual methods should be created for each menu item. This will allow the user to recall a method in the future if needed.
- Signed Representation should be included in the program, allowing for positive or negative numbers.

Mathematics for Computing and IT:

- Produce a typed report and saved as a PDF
- Report must answer all of the questions outlined in the mathematics rubric below.
- Name, Student Number, Group clearly stated at the top of your report
- All content researched and clearly referenced.

Computer Programming:

- A single .java source code for the application with comments outlining each individual method that is created.
- All code should be developed inside of a **single file**. Additional files are not needed for this program.
- Comments added to the top of the source code file outlining your name, student number and class group (A or B).

Notes

- This is an individual assignment, plagiarism will not be tolerated.
- Usual late penalties apply, 5% for each day late up to a maximum of 5 days, after which a result of 0% will be recorded.
- Submissions are **not accepted by email**. All submissions must be made through the Moodle upload links.
- All work that is submitted must be your own. This is not a group assignment. Any work resembling another student's work will be given a zero grade for both involved.

Mathematics for Computing & IT

- Your report must be submitted as a single PDF document, typed. All graphs must be generated.
- Output should be a screen shot.

Computer Programming

- A single .java file should be uploaded to the Moodle uploader link provided. Please ensure you **do not** upload the class file as these are not human readable.
- Do not zip the file.

Marking Scheme Summary

Mathematics for Computing and IT

Description	Weighting
<p>[Logic & Program Flow 1]</p> <ol style="list-style-type: none"> Outline the structure of your program using an object diagram and an overview of the program progression using flow chart. Describe the process of the menu system, using a flow chart. Translate your flow chart to pseudo code, using method names and selection statements where appropriate. 	<p>[0 – 10]</p> <p>[0 – 5]</p> <p>[0 – 5]</p>
<p>[Logic & Program Flow 2]</p> <ol style="list-style-type: none"> Give an example of a Propositional Logical Statement from your pseudo code above (give the line of code and explain its meaning). Describe the code in terms of a worded logical statement (English sentence). Translate the worded logical statement into primitive propositions and logical operators and hence a mathematical logical compound proposition. Give a full truth table for this proposition. Research Conjunctive Normal Form (CNF). Define CNF and then translate your mathematical logical expression into its conjunctive normal form, showing your workings. Research First Order Logic (FOL). Define FOL and then give an example from your code and explain using a truth table. 	<p>[0 – 30]</p> <p>[0 – 10]</p> <p>[0 – 10]</p> <p>[0 – 10]</p>
<p>[Number Systems 1 – Binary to Decimal Conversion]</p> <ol style="list-style-type: none"> Describe mathematically the <u>Binary to Decimal Conversion</u> process. Explain how you would implement this into a program. Provide pseudo code for this method and explain. Provide an iterative table outlining the steps taken to convert the binary number 10100 to decimal. 	<p>[0 – 20]</p> <p>[0 – 5]</p> <p>[0 – 5]</p> <p>[0 – 10]</p>
<p>[Number Systems 2 – Signed Representation]</p> <ol style="list-style-type: none"> Describe mathematically <u>Signed Representation</u> for negative numbers using the <u>Two's Complement</u> process. Explain how you would implement this into a program. Provide pseudo code for this method and explain. Provide an iterative table outlining the steps taken to convert the 	<p>[0 – 20]</p> <p>[0 – 5]</p> <p>[0 – 5]</p>

binary number -56 to a signed binary number.	[0 – 10]
[Output: For each of the following provide a <u>separate</u> screenshot of the output clearly indicating the answer] 12. Convert 345 & 98.15 to binary. Convert 10110011 & 11011010.1010 to decimal 13. Represent the following in signed representations: 23, -54, 156, -234 14. Using binary arithmetic: Add (28 + 34), Subtract (– 134 – 56), Multiply (45 * 23), Divide (40.25 / 5.5)	[0 – 15] [0 – 2.5] [0 – 2.5] [0 – 10]
[Presentation of Report] 15. Adhere to Specifications & Guidelines, Layout, Format, Innovation	[0 – 5] [0 – 5]
[TOTAL]	[0 - 100]

Computer Programming

Description	Weighting 100%
Binary to decimal method functioning correctly, storing data inside global variable that can be accessed later by math methods.	0-10
Decimal to binary method function correctly, storing data inside global variable that can be accessed later by math methods.	0-10
Menu system presented to the user, allowing the user to navigate into each individual program method. When the user has finished, the will be returned to the main menu allowing them to perform the next operation.	0-5
Methods utilising return statements and passing variables to methods.	0-5
Addition function working correctly on defined user input variables (Variable 1 and Variable 2)	0-10
Subtraction function working correctly on defined user input variables (Variable 1 and Variable 2)	0-10
Multiplication function working correctly on defined user input variables (Variable 1 and Variable 2)	0-10
Division function working correctly on defined user input variables (Variable 1 and Variable 2)	0-10
Signed Representation of numbers functioning correctly and working with each individual method.	0-10
Robustness of data inputted by the user (validating for blanks, incorrect values being entered, length)	0-10
Overall code quality, style and commenting	0-10
Total	100%

Sample README File

Name: my name

Student number: 123456

My code does the following:

- file 1 does ...

- file 2 does ...

- ...

To compile my code:

do the following steps

To run my code:

do the following steps