



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

LEVEL/ LEFEL : 4

MODULE CODE/ COD Y MODIWL: ACCC4006
SOFTWARE DEVELOPMENT

ASSESSMENT TYPE/ MATH O ASESU: ASSIGNMENT

ASSESSMENT/COMPONENT/ ASESAD/CYDRAN: 1 & 2

COMPONENT WEIGHTING/ PWYSAU'R CYDRAN: 100%

WORD COUNT / NIFER Y GEIRIAU: 4000 WORDS/O EIRIAU

SUBMISSION DETAILS/ MANYLION CYFLWYNO:

- **Make sure that Your Name, Your Student Number, Your Module Title, Assignment Title and Your Module Lecturer's Name are clearly shown on the front page of your assignment**
- **All assignments must be submitted electronically to Moodle.**
- **DO NOT put this form into Turnitin or it will match many similarities with other students' submissions.**

School of Applied Computing ASSIGNMENT SPECIFICATION

Programme:	BSc (Hons) Computer Networks / Cloud Computing		
Assignment Title:	Software Development		
Hand Out Date:	The 3 rd Week of Term	Submission deadline:	Please refer to the assessments schedule published on Students' Hall in Moodle and the Assessment Board on campus

Referencing:	In the main body of your submission you must give credit to authors on whose research your work is based. Append to your submission a reference list that indicates the books, articles, etc. that you have read or quoted in order to complete this assignment (e.g. for books: surname of author and initials, year of publication, <u>title of book</u> , edition, publisher: place of publication).
--------------	---

Late submission will result in a late penalty mark, as follows:

**Up to one week late, maximum mark of 40% for first attempts and 0% for resubmissions. No work will be accepted more than one week after the submission deadline
(Academic Quality Handbook 2016/17 7.5 (5))**

AIM(S)

To provide the student with the essential software concepts involved in the implementation, testing and debugging of programs. Simple design concepts and good programming practices will be adopted throughout the module.

LEARNING OUTCOMES

Upon the successful completion of this module, the student should be able to demonstrate the ability to:

1. Demonstrate an understanding of programming by analysing and developing solutions to a variety of problems, using software libraries/frameworks if and where appropriate.
2. Determine appropriate fundamental concepts and techniques that can be deployed in the solution of problems/scenarios.

ASSIGNMENT

COMPONENT: 1 + 2 (50% Each)

The assignment comes in two sections, each section is worth 50% of the overall marks for the module and the results for each section will be recorded separately (out of 100%)

COMPONENT WEIGHTING: 100 %

SUBMISSION DETAILS:

As this course is assessed by assignment only, a high quality deliverable is expected.

Check List

- 1) Create a main folder and name it with your **student's ID**. Create another two folders and name them as Assignment1_ **student's ID** and Assignment2_ **student's ID**. Save your programs into the corresponding folders. Written report answering the questions and describing the code, should be named as REPORT1_ **student's ID** for first section and REPORT2_ **student's ID** for second section and save the reports in corresponding folders. Zip your main folder and upload it on Moodle using advance upload.

FAILURE TO MEET THE ABOVE REQUIREMENTS MAY RESULT IN THE STUDENTS WORK NOT BEING MARKED.

Title page: Including author name and affiliation (relationship with project).

Sections: Ensure a TOC 'Table of contents' page is provided at the beginning of the report listing all report sections.

Numbering: Each report section including main headings and sub-headings should be numbered throughout.

All pages should be numbered, Arabic numerals (1, 2, 3, etc.) on all pages consecutively throughout the text, appendices, and bibliography.

Consistency: A document is consistent if, for example, it always uses the same type style including face and size for document objects such as headings, sub-headings, references and content text and all lines should have the same spacing etc.

Grammar: Technical documents use 'passive voice' expressions, for example your report should avoid using the word 'I' as much as possible.

TASK DESCRIPTION

This assignment is designed to test your understanding of the coding concepts covered in class, your ability to read & interpret code, your ability to test code and your ability to develop your own code.

The assignment comes in two sections, each section is worth 50% of the overall marks for the module and the results for each section will be recorded separately (out of 100%)

You are advised to complete part A of both sections 1 and 2 first.

SECTION 1 – 50%

In a written report (2000 words) complete the following:

PART A (UNDERSTANDING)

1. Explain the difference between setting a variable to a constant number and using the keyword CONST to define a constant in c#.
2. Explain data types and mention why these are necessary? With the use of suitable examples, discuss how to convert from one type to another (include both implicit and explicit type conversion).
3. Explain the different types of control structures in programming. With reference to repetition, compare and contrast the different types of loops in programming.

PART B (INTERPRETATION)

The source code for an application has been provided in the Appendix A.

Run the code and **describe how the application works**, creating step by step instructions for using this application. Preferably, illustrate your answer with appropriate screen shots and diagrams e.g. flow charts.

Read and **provide a line-by –line explanation of how the code** for this application has been implemented. Take care to include a description of the concepts used including the arguments being passed, return types, etc.

PART C (IMPLEMENTATION)

FULL PRINTED CODE LISTINGS MUST BE PROVIDED FOR MARKING AND SCREEN SHOTS SHOULD ALSO BE TAKEN WHERE APPROPRIATE

YOU DO NEED TO COMPLETE ALL THE EXERCISES TO PASS THIS ASSIGNMENT, AND YOU MAY JUST ATTEMPT ELEMENTS OF AN EXERCISE. THE FOLLOWING EXERCISES BECOME INCREASINGLY MORE CHALLENGING, THE OBJECTIVE IS TO DEVELOP A SYSTEMATIC APPROACH TO SOLVING PROBLEMS

Please Note:

This assignment is designed to test your problem solving skills, thus marks will be awarded for:

- Identifying specific problems in completing the exercise.
 - You may also constrain a problem (simplify it) to produce a simpler solution that may act as a stepping stone to solving the real problem

- Demonstrating solutions to these problems.
- A clear analysis of the solutions (the thought process behind them)
- Testing that demonstrates clearly that the solutions works under all circumstances

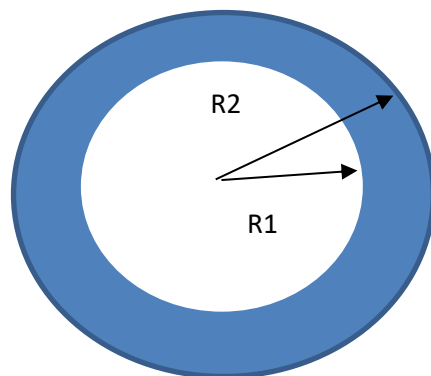
Exercise 1:

Design and construct a program to calculate average marks for a student on a course. Your program should allow the user to declare different types of variables to hold name of the module, mark for a module. Some of the modules could be:

- Academic and Employability Skills
- Computational Thinking
- Computer Architecture
- Information Engineering
- Internet Protocols
- IP Internetworking
- Network Principles
- Software Development

Exercise 2:

Design and construct a program to allow the user to enter the radii to calculate the shaded area.



Formula for Area of Circle:

$$A = \pi r^2$$

Figure 1

Exercise 3:

Construct a program to enter a list of names of pupils in a class and their corresponding marks for coursework and examination. The coursework and examination is weighted at 50% of the module mark. There are some regulations surrounding the marking, namely:

For the student to pass the module, they must have an overall average mark between coursework and examination of 40%. However, they must also have a minimum mark of 40% in each component (i.e. coursework and examination);

TASKS:

1. Construct a flowchart and program to enter the student details:
 - a. Name;
 - b. Coursework Mark
 - c. Examination Mark

2. Having entered the data above, your program should calculate the module mark, highlighting those students who have not met the pass mark, either individually in either component or overall.
3. Your program should produce the following reports on-screen:
 - a. The results list, including the average mark for the module and the standard deviation;
 - b. List all those students who have not met the criteria and require further discussion at the examination board;
 - c. A graduated pass list indicating the grade of each student i.e. 1st Class, Upper 2nd Class, Lower 2nd Class, Third Class and Fail.
4. Your program should make use of suitable Methods (functions) where appropriate.

SECTION 2 – 50%

In a written report (2000 words) complete the following:

PART A (UNDERSTANDING)

1. Define Arrays and explain the importance of arrays in programming. Give relevant examples to support your answer.
2. In OOP define inheritance, use suitable coding examples to demonstrate your answer.

PART B (TESTING)

Write a short essay on testing, selecting at most 3, explaining each testing model, and comparing and contrasting each. Your essay should not exceed 200 words.

PART C (IMPLEMENTATION)

FULL PRINTED CODE LISTINGS MUST BE PROVIDED FOR MARKING AND SCREEN SHOTS SHOULD ALSO BE TAKEN WHERE APPROPRIATE

YOU DO NEED TO COMPLETE ALL THE EXERCISES TO PASS THIS ASSIGNMENT, AND YOU MAY JUST ATTEMPT ELEMENTS OF AN EXERCISE. THE FOLLOWING EXERCISES BECOME INCREASINGLY MORE CHALLENGING, THE OBJECTIVE IS TO DEVELOP A SYSTEMATIC APPROACH TO SOLVING PROBLEMS

Please Note:

This assignment is designed to test your problem-solving skills, thus marks will be awarded for:

- Identifying specific problems in completing the exercise.
 - You may also constrain a problem (simplify it) to produce a simpler solution that may act as a stepping stone to solving the real problem
- Demonstrating solutions to these problems.
- A clear analysis of the solutions (the thought process behind them)
- Testing that demonstrates clearly that the solutions works under all circumstances

Programming Exercise

Construct a program using OOP concepts, for the purpose representing a car. You should make use of classes and methods to allow the car to be constructed and represented in the computer. The car should have the following data to describe it as a minimum:

1. VRN – Vehicle Registration Number e.g. XY 20 VGD
2. Manufacturer e.g. Ford
3. Model: Kuga
4. Engine Size: e.g. 1500cc
5. No of Seats: 5
6. Type: Saloon/Hatchback/SUV/4x4/Coupe
7. Max Speed: e.g. 120mph
8. Power: 106bhp;
9. Current speed.

Your class should contain some methods to allow for:

1. Forwards;
2. Backwards;
3. Rough Terrain for 4x4 or 4 wheel drive vehicles;
4. Set the speed of the vehicle.

Run the program with test data to show that your program can create a number of vehicles, showing the direction and speed.

CRITERIA SECTION 1

Research-Informed Literature

PART A	%Max
Knowledge of Variables	5
Knowledge of Casting	5
Knowledge of Iteration	10

Knowledge and Understanding of Subject

PART B	%Max
Run and Document The Application	15

Analysis

PART B	%Max
Description of the code	15

Practical Application and Deployment

PART C	%Max
Exercise 1	5
Exercise 2	10
Exercise 3	10
Exercise 4	25

MARKING CRITERIA SECTION 2

Research-Informed Literature

PART A	%Max
Knowledge of Arrays	20
Knowledge of Inheritance and OOP Concepts	20

Analysis

PART B	%Max
Test Report on Testing Schemes	10

Practical Application and Deployment

PART C	%Max
Creation of Classes	10
Demonstrate Class implementation and Inheritance	10
Public Void Main Program Implementation	20

COMMON ASSESSMENT AND MARKING CRITERIA

	OUTRIGHT FAIL	UNSATISFACTORY	SATISFACTORY	GOOD	VERY GOOD	EXCELLENT	EXCEPTIONAL
Assessment Criteria	0-29%	30-39%*	40-49%	50-59%	60-69%	70-79%	80-100%
1. Research-informed Literature Extent of research and/or own reading, selection of credible sources, application of appropriate referencing conventions	Little or no evidence of reading. Views and findings unsupported and non-authoritative. Referencing conventions largely ignored.	Poor evidence of reading and/or of reliance on inappropriate sources, and/or indiscriminate use of sources. Referencing conventions used inconsistently.	References to a limited range of mostly relevant sources. Some omissions and minor errors. Referencing conventions evident though not always applied consistently.	Inclusion of a range of research-informed literature, including sources retrieved independently. Referencing conventions mostly consistently applied.	Inclusion of a wide range of research-informed literature, including sources retrieved independently. Selection of relevant and credible sources. Very good use of referencing conventions, consistently applied.	A comprehensive range of research informed literature embedded in the work. Excellent selection of relevant and credible sources. High-level referencing skills, consistently applied.	Outstanding knowledge of research-informed literature embedded in the work. Outstanding selection of relevant and credible sources. High-level referencing skills consistently and professionally applied.
2. Knowledge and Understanding of Subject Extent of knowledge and understanding of concepts and underlying principles associated with the discipline.	Major gaps in knowledge and understanding of material at this level. Substantial inaccuracies.	Gaps in knowledge, with only superficial understanding. Some significant inaccuracies.	Evidence of basic knowledge and understanding of the relevant concepts and underlying principles.	Knowledge is accurate with a good understanding of the field of study.	Knowledge is extensive. Exhibits understanding of the breadth and depth of established views.	Excellent knowledge and understanding of the main concepts and key theories. Clear awareness of challenges to established views and the limitations of the knowledge base.	Highly detailed knowledge and understanding of the main theories/concepts, and a critical awareness of the ambiguities and limitations of knowledge.
3. Analysis Analysis, evaluation and synthesis; logic, argument and judgement; analytical reflection; organisation of ideas and evidence	Unsubstantiated generalisations, made without use of any credible evidence. Lack of logic, leading to unsupportable/ missing conclusions. Lack of any attempt to analyse, synthesise or evaluate.	Some evidence of analytical intellectual skills, but for the most part descriptive. Ideas/findings sometimes illogical and contradictory. Generalised statements made with scant evidence. Conclusions lack relevance.	Evidence of some logical, analytical thinking and some attempts to synthesise, albeit with some weaknesses. Some evidence to support findings/ views, but evidence not consistently interpreted. Some relevant conclusions and recommendations, where relevant	Evidence of some logical, analytical thinking and synthesis. Can analyse new and/or abstract data and situations without guidance. An emerging awareness of different stances and ability to use evidence to support the argument. Valid conclusions and recommendations, where relevant	Sound, logical, analytical thinking; synthesis and evaluation. Ability to devise and sustain persuasive arguments, and to review the reliability, validity & significance of evidence. Ability to communicate ideas and evidence accurately and convincingly. Strong, convincing conclusions / recommendations.	Thoroughly logical work, supported by evaluated evidence. High quality analysis, developed independently or through effective collaboration. Ability to investigate contradictory information and identify reasons for contradictions. Strong, persuasive, conclusions, justifiable recommendations.	Exceptional work; judiciously selected and evaluated evidence. Very high quality analysis, developed independently or through effective collaboration. Ability to investigate contradictory information and identify reasons for contradictions. Highly persuasive conclusions
4. Practical Application and Deployment Effective deployment of appropriate methods, materials, tools and techniques; extent of skill demonstrated in the application of concepts to a variety of processes and/or contexts; formulation of innovative and creative solutions to solve problems.	Limited or no use of methods, materials, tools and/or techniques. Little or no appreciation of the context of the application.	Rudimentary application of methods, materials, tools and/or techniques but without consideration and competence. Flawed appreciation of the context of the application.	An adequate awareness and mostly appropriate application of well established methods, materials, tools and/or techniques. Basic appreciation of the context of the application.	A good and appropriate application of standard methods, materials, tools and/or techniques. Good appreciation of the context of the application, with some use of examples, where relevant.	A very good application of a range of methods, materials, tools and/or techniques. Very good consideration of the context of the application, with perceptive use of examples, where relevant. Evidence of some innovation and creativity.	An advanced application of a range of methods, materials, tools and/or techniques. The context of the application is well considered, with extensive use of relevant examples. Application and deployment extend beyond established conventions. Innovation and creativity evident throughout.	Outstanding levels of application and deployment skills. Assimilation and development of cutting edge processes and techniques.
5. Skills for Professional Practice Demonstrates attributes expected in professional practice including: individual initiative and collaborative working; deployment of appropriate media to communicate (including written and oral); clarity and effectiveness in presentation and organisation.	Communication media is inappropriate or misapplied. Little or no evidence of autonomy in the completion of tasks. Work is poorly structured and/or largely incoherent.	Media is poorly designed and/or not suitable for the audience. Poor independent or collaborative initiative. Work lacks structure, organisation, and/or coherence	Can communicate in a suitable format but with some room for improvement. Can work as part of a team, but with limited involvement in group activities. Work lacks coherence in places and could be better structured.	Can communicate effectively in a suitable format, but may have minor errors. Can work effectively as part of a team, with clear contribution to group activities. Mostly coherent work and is in a suitable structure.	Can communicate well, confidently and consistently in a suitable format. Can work very well as part of a team, with very good contribution to group activities. Work is coherent and fluent and is well structured and organised.	Can communicate professionally and, confidently in a suitable format. Can work professionally within a team, showing leadership skills as appropriate, managing conflict and meeting obligations. Work is coherent, very fluent and is presented professionally.	Can communicate with an exceptionally high level of professionalism. Can work exceptionally well and professionally within a team, showing advanced leadership skills. Work is exceptionally coherent, very fluent and is presented professionally.

**PAGE INTENTIONALLY
LEFT BLANK**

APPENDIX A:

```
using System;

namespace Glazier_App
{
    class Program
    {
        static void Main(string[] args)
        {
            double width, height, woodLength, glassArea;
            const double MAX_WIDTH = 5.0;
            const double MIN_WIDTH = 0.5;
            const double MAX_HEIGHT = 3.0;
            const double MIN_HEIGHT = 0.75;
            string widthString, heightString;

            do
            {
                Console.WriteLine("Give the width of the window between " + MIN_WIDTH + " and " + MAX_WIDTH + " : ");
                widthString = Console.ReadLine();
                width = double.Parse(widthString);
            } while (width < MIN_WIDTH || width > MAX_WIDTH);

            do
            {
                Console.WriteLine("Give the height of the window between " + MIN_HEIGHT + " and " + MAX_HEIGHT + " : ");
                heightString = Console.ReadLine();
                height = double.Parse(heightString);
            } while (height < MIN_HEIGHT || height > MAX_HEIGHT);

            woodLength = 2 * (width + height) * 3.25;

            glassArea = 2 * (width * height);

            Console.WriteLine("The leng of the wood is " + woodLength + " feet");
            Console.WriteLine("The area of the glass is " + glassArea + " Square metres");

        }
    }
}
```