COMP3002
Alternative Computing Paradigms

20 CREDIT MODULE

ASSESSMENT: 100% Coursework **W1: 30% Set Exercises**
W2: 70% Report

MODULE LEADER: Thomas Wennekers

MODULE AIMS

- To expose students to ways of thinking about computational problems different from mainstream imperative styles.
- To train students in alternative computing paradigms like declarative or functional programming
- To widen students' perspective on computing by introducing them to state-of-the-art novel technology like quantum computing or neural computing

ASSESSED LEARNING OUTCOMES (ALO):

1. Critically evaluate the appropriateness of computing paradigms for a particular application.
2. Apply the use of an alternative paradigm (e.g. functional programming) to produce a solution to a problem.
3. Analyse a technology at the forefront of computational science.

Overview

This document contains all the necessary information pertaining to the assessment of *COMP3002 Alternative Computing Paradigms*. The module is assessed via **100% coursework**, across two elements: *30% Set Exercises* and *70% Report*.

The sections that follow will detail the assessment tasks that are to be undertaken. The submission and expected feedback dates are presented in Table 1. All assessments are to be submitted electronically via the respective DLE module pages before the stated deadlines.

	Submission Deadline	Feedback
Set Exercises (30%)	**27/03/23 4pm**	Within 20 working days
Report (70%)	**16/05/23 4pm**	Within 20 working days

Table 1: Assessment Deadlines

All assessments will be introduced in class to provide further clarity over what is expected and how you can access support and formative feedback prior to submission. Whilst the assessment information is provided at the start of the module, it is not necessarily expected you will start this immediately – as you will often not have sufficient understanding of the topic. The module leader will provide guidance in this respect.

Assessment 1: Set Exercises

Solve the subsequent tasks. Hand in solutions as a single Word or PDF file. Where code is requested, it should be proper and correct Python code copied into the file as plain text or a screenshot. Make sure the code is well readable. Explain your solutions well. Don't just rely on Wikipedia or similar resources. These are often very short and sometimes wrong.

Tasks:

Q1 Recursion and Dynamic Programming in Python

[Total: 25 marks]

a) Explain, using the data-structure of a graph as an example, what is meant by a recursive data type and describe three others.

[2 marks]

b) Explain what is meant by single and multiple function recursion and give examples using the factorial function and the Fibonacci numbers.

[2 marks]

c) Explain what is meant by tail recursion and why it theoretically allows to write more efficient code as compared to general recursion. Why does tail recursion not work efficiently in Python?

[2 marks]

d) Explain what is meant by a divide-and-conquer algorithm. Explain in detail how one could use the divide-and-conquer principle to sort a set of integer numbers by the product of their digits. You don't need to provide code.

[2 marks]

e) Explain in detail how a general recursive function can be processed using a run-time stack.

[4 marks]

f) Write a Python function to calculate the greatest common divisor of a list of numbers. Avoid loops if you can. Explain in detail how your solution works.

[5 marks]

g) Write a Python class that finds the longest common substring of a set of strings. Use memoization. Use comments in the code to explain how your code works.

[8 marks]

Q2: Randomized Algorithms

[Total 40 Marks]

a) Assume you have a biased random bit generator (in Python) called biased_bits that when called returns bit 1 with probably $0 < p < 0.5$ and a 0-bit else. Using biased_bits, write a

function random_float, that generates floating point numbers equally distributed on the interval [0,1[. Explain why and how your solution works.

[5 marks]

b) Write a function that returns a random node of a linked list such that any node has the same probability of being returned.

[15 marks]

c) Write a function that returns a random node of a tree (not necessarily binary) such that any node has the same probability of being returned.

[20 marks]

The functions in a-c should be computationally as efficient (fast) as possible. This may require some memory overhead. The number of nodes (N) in cases b and c may be unknown (you may want to write two functions, one where N is known and one where it is not). Explain how and why your solutions work, and how efficient they are in space and time using O-Notation.

Q3 Functional Programming in Python

[Total 20 marks]

a) Explain the main idea behind functional programming by contrasting it with Object-Orientated Programming, as well as its advantages and disadvantages.

[3 marks]

b) Explain using Python examples what higher order functions are and what “functions are treated as first-class citizens” means.

[3 marks]

c) Explain the concepts, commonalities and differences of generators, iterators and closures using Python examples.

[3 marks]

d) Write a function called ‘compress_keys’ that takes a dictionary with string keys and returns a new dictionary with the vowels removed from the keys. The function should use a list comprehension nested inside a dict comprehension.

[2 marks]

e) Write a Python function that maps a lambda expression for the square-root function to a list of integers.

[2 marks]

f) Write a Python decorator that wraps the function from part e) and checks if any of the arguments in the list is smaller or equal to zero. If so, an error message should be printed. This should use a comprehension.

[2 marks]

g) Explain the concept of Monads as well as their advantages.

[5 marks]

Assessment Criteria: Correctness and completeness of solutions, completeness and clarity of explanations, good programming practice.

Threshold Criteria (these are indicative only):

To achieve a pass (40%+), 40% of the total marks of this assessment element.
To achieve a merit (60%+), 60% of the total marks of this assessment element.
To achieve a distinction (70%+), 70% of the total marks of this assessment element.

Assessment 2: Report

Task:

This course work part consists of research into a topic related to alternative computing paradigms written up in form of an individual scientific report of 2,000 words. Topics to choose from are listed further down the page.

The uploaded report should be a pdf file. Microsoft Word .doc or .docx files, or .odt files will be accepted too, but often display odd formatting for various technical reasons. This could affect marks.

The report must be zipped up with annotated pdfs of the three most important resources used (usually scientific publications) and the zip-file be uploaded via the submission link on the module's DLE pages.

Aim

You are expected to carry out research into an area of Alternative Computing Paradigms and prepare a *scientific report* about your findings. The aim of the report is to allow you to demonstrate your ability to investigate in depth and present in writing an aspect of the area of Alternative Computing Paradigms. A scientific report should be more formal and go into more depth than an essay.

Content

The work you will carry out must be in one of the following areas:

1) Non-Boolean Logic: for example, an in-depth exploration of any of first-order logic, modal logic, fuzzy logic, probabilistic logic, multi-valued logic, or quantum logic. These alternative logic theories extend the concept of Boolean logic (the one you know) in various ways. The list is not exhaustive, there are more alternative forms of logics you could choose from.

2) Simulation: Computer Simulations are used in many different areas of Science, Research and Development, for example, in game engines; the simulation of physical products like car parts, ovens, or chemical or power plants; the generation of Computer music using the physical modeling of string or pipe instruments; prediction of the stock market, and many more areas. The goal is usually to predict properties of the simulated object or systems. You can choose one area of Simulation and explore how Computer Science is used to benefit that domain of research or development.

3) SpiNNaker and Spiking Neurons: SpiNNaker is a neuro-inspired hardware technology currently developed by Steve Furber's group at the University of Manchester. They aim at simulating (mouse) brains in real-time within the next 10 years. Find out what might be possible with this new technology and how it works.

Some of these topics are quite broad and may therefore be difficult to cover in the short space given. You are allowed to constrain them to a more specific sub-topic that can be covered within about 50-70 hours work and within the available space. For this, you would read into the general topics to gain some overview into what they are about and then choose a more constraint topic of your liking to dive into it in depth. If in doubt about your topic of choice, ask the module team.

There are few additional constraints other than that your work must have a clear link to Computing/Computer Science. Pure Physics, Mathematics, Engineering, Philosophy etc. would be inappropriate. However, depending on your topic you may need some interdisciplinary content from these areas of science.

You should read scientific literature about your topic of choice, not just Blog-, News-, Company- or Amateur-webpages about the issues. Scientific journals provide resources of higher quality. You are expected to read and use at least three scientific publications. The manuscripts of three publications must be included in your submission as annotated PDFs (using for example Adobe Reader). The annotations should reflect the level of depth to which you have read the papers. It may be difficult to find scientific papers for some more applied topics; in this case use the best material you can find.

After your research, you will write a scientific report similar in form to scientific reviews or journal papers (compare how the papers you read are composed). The report should have figures to explain concepts and results, and references cited in the text in Harvard style.

One of the lectures will be used to explain the coursework assessment in detail.

Marking scheme

The report should have the form of a scientific review or research paper with the following components (percentages are marks out of 100%).

1. 10% Introduction – what is the investigation about and why is it of interest.
2. 20% Background and methods: Description of the fundamental concepts and ideas necessary to understand section 3.
3. 50% Presentation of some specific main issues on the topic.
4. 10% Summary, discussion, and conclusions – what are the main points that can be learnt from the investigation and what are problems, that still need to be understood or worked out?
5. 10% Background literature – what are the sources of information for your investigation (journal papers, books, websites)?

Marks will be awarded based for:

1. The organisation and clarity of your written presentation
2. Your use of figures and diagrams where helpful in illustrating key points
3. The breadth and depth of your investigation
4. The extent to which you can constructively bring together various aspects of the topic you are investigating to provide greater insight/understanding for the reader.

5. Your ability to summarise the key ideas at the end of your report.
6. The quality of your referencing and background literature

Threshold Criteria (these are indicative only):

Pass mark is 40 percent: to pass requires some solid research using at least three reasonable resources or publications documented in a completed and mostly error-free report. To achieve a merit (60%) the report must well written, clearly be built on high-quality scientific sources, and reflect a very good overview and understanding. Distinction level is 70 percent and requires an excellent overview over the chosen field, an in-depth understanding of the issues discussed, and the explicit use of high-quality academic resources.

Introduction	/10
Background and Methods	/20
Main Issues	/50
Summary, Discussion and Conclusions	/10
References (and citations)	/10
Overall	/100

Feedback Template for Assessment 2

General Guidance

Extenuating Circumstances

There may be a time during this module where you experience a serious situation which has a significant impact on your ability to complete the assessments. The definition of these can be found in the University Policy on Extenuating Circumstances here: https://www.plymouth.ac.uk/uploads/production/document/path/7/7741/Extenuating_Circumstances_Policy_and_Procedures.pdf

Students with valid EC's may claim either for an extension period or for non-submission. Valid for Non-Submission means you will be asked to do a new piece of work. Please see below.

Referral

Please note that if you claim for non-submission and are offered a referral, you will be required to complete a NEW piece of work for the module. The new piece of work will assess that you have met the learning outcomes for the module but in a way that will be different to the original set piece. The referral is not a repeat or extension of the original coursework.

Carrying out a new piece of work means you will not be able to keep the marks already gained during the module. Eg: if you pass the set exercises (CW1) but do not submit the main work (CW2) AND are offered a referral, you will not keep the set exercises grade.

Plagiarism

All of your work must be of your own words. You must use references for your sources, however you acquire them. Where you wish to use quotations, these must be a very minor part of your overall work.

To copy another person's work is viewed as plagiarism and is not allowed. Any issues of plagiarism and any form of academic dishonesty are treated very seriously. All your work must be your own and other sources must be identified as being theirs, not yours. The copying of another persons' work could result in a penalty being invoked.

Further information on plagiarism policy can be found here:

Plagiarism: https://www.plymouth.ac.uk/student-life/your-studies/essential-information/regulations/plagiarism

Examination Offences: https://www.plymouth.ac.uk/student-life/your-studies/essential-information/exams/exam-rules-and-regulations/examination-offences

Turnitin (http://www.turnitinuk.com/) is an Internet-based 'originality checking tool' which allows documents to be compared with content on the Internet, in journals and in an archive of previously submitted works. It can help to detect unintentional or deliberate plagiarism.

It is a formative tool that makes it easy for students to review their citations and referencing as an aid to learning good academic practice. Turnitin produces an 'originality report' to help guide you. To learn more about Turnitin go to:
https://guides.turnitin.com/01_Manuals_and_Guides/Student/Student_User_Manual

Referencing

The University of Plymouth Library has produced an online support referencing guide which is available here: http://plymouth.libguides.com/referencing.

Another recommended referencing resource is Cite Them Right Online; this is an online resource which provides you with specific guidance about how to reference lots of different types of materials.

The Learn Higher Network has also provided a number of documents to support students with referencing:
References and Bibliographies Booklet:
http://www.learnhigher.ac.uk/writing-for-university/referencing/references-and-bibliographies-booklet/
Checking your assignments' references:
http://www.learnhigher.ac.uk/writing-for-university/academic-writing/checking-your-assigments-references/