PROGRAM DESIGN

WEEK 1: MODULE OVERVIEW

INTRODUCTION

- **About Jonathan:**
- Jonathan McCarthy
- Lecturer – School of Computing

- **Contact Details:**
- If you have any questions or queries regarding the Program Design module please send me a mail: **jonathan.mccarthy@dit.ie**

MODULE TIMETABLE

Course schedule

- 2 hours lectures
  - Monday 10:00-11:00 (KE G-007)
  - Monday 12:00-13:00 (KE G-007)
- 1 hour Tutorial
  - Monday 16:00-17:00 (KE 4-008)
- 1 hour lab
  - Tuesday 11:00-12:00

MODULE NOTES

- All course content will be available through Webcourses.
- Self Enrol Key (DT228): coffee
- Self Enrol Key (DT282): orange

- If you cannot access the content please email Jonathan asap!!
- **jonathan.mccarthy@dit.ie**

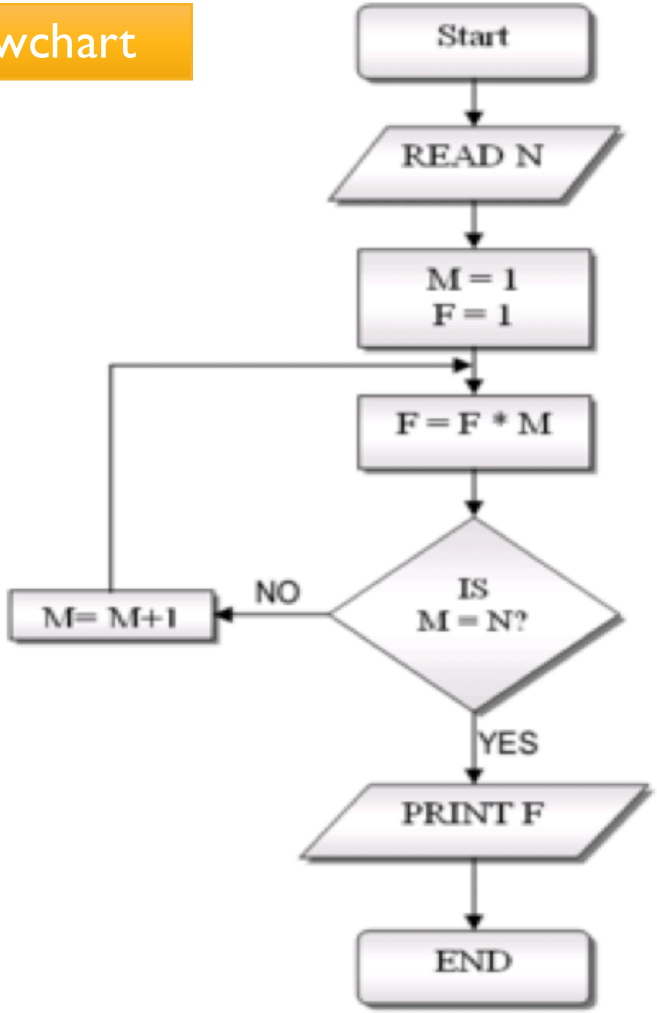
MODULE DESCRIPTION

This module is concerned with

- **Program design skills**
  - Flowcharts
  - Pseudocode
  - and programming language constructs which are used to model and design computer programs.
- Consideration is given as to how **problem** information might be **represented** in **code** or on **paper.**
- And what **program design steps** may be performed to arrive at a solution.
- **Abstraction**, **modularity** and **top-down** design

EXAMPLE OF FLOWCHART AND PSEUDOCODE

flowchart



Pseudocode

```
Set total to zero

Set grade counter to one

While grade counter is less than or equal to ten
    Input the next grade
    Add the grade into the total

Set the class average to the total divided by ten

Print the class average.
```

MODULE AIMS

Introduction to:

- Program **design techniques**
- Design **strategies**
  - top-down
  - bottom-up
  - stepwise refinement.
- The use of pseudocode and flowcharts in program design.
- **Convey**
  - An understanding and appreciation of the **power of abstraction**
  - whereby the essential information relating to a programming problem is abstracted and **mapped onto programming constructs.**
- **The importance of a well conceived design before rushing into code.**

LEARNING OUTCOMES

On completion you should be able to

- **Abstract problem information** and represent it on paper or an appropriate computing environment.
- **Demonstrate** a basic competence in the **use of a program constructs** to solve a problem
- **Develop solutions** to some elementary program design problems using **top down design and stepwise refinement.**
- **Describe** some simple program designs **using pseudocode and flowcharts**, and then implement the design.

MODULE CONTENT

- Problem Solving, Stages in Problem Solving.
- Data Types and Data Representation.
- Program Constructs.
- Abstraction, Problem Specification, Approaches to Problem Solving and Program Construction, Divide and Conquer, Stepwise Refinement, Top Down Design, Bottom Up Design.
- Recursion.
- Greatest Common Divisor, Factorial and Fibonacci.
- Pseudocode and Flowcharts in Program Design.
- Linear Data Structures - arrays and lists.

MODULE ASSESSMENT

- Assessment will be based on a two hour end of semester written exam and continuous assessment during the semester.

- **Written exam - 60%**

- **Continuous Assessment - 40%**
  - Labs – 10% (Lab work for weeks 1 to 5) Each lab carries equal marks
  - Mid module assessment released – 30%

LATE SUBMISSIONS

- **Rules for late submissions:**
- **Week 1:** 4% for the first day, 1% for each day thereafter.
- **Week 2:** 2% for each day thereafter.
- **Week 3:** No submissions accepted, zero grade.

- **Note:** All penalties are calculated per day started.

READING MATERIAL

- No specific textbook.
- Computer Science is almost unique as a discipline in the sense that are a massive number of computer books online for free, as many leading computer science authors release their books on-line as "open source" and there are a number of sites that collect these books;
- http://freecomputerbooks.com/
- http://www.freebookcentre.net/
- http://www.intelligentedu.com/free_computer_books.html
- http://www.onlinecomputerbooks.com/
- http://www.computer-books.us/