PROGRAM DESIGN
1. MODULE OVERVIEW

Contact the lecturer

- Dr. Qi Wang
- Email: qi.wang@dit.ie
- Office: KE-1-007A
- Phone: 01-4024852

Module Timetable

- Lectures (2 hours)
  - Monday 12:00–13:00 (KE-G-007)
  - Wednesday 10:00–11:00 (KE-3-008)

- Tutorial (1 hour)
  - Thursday 11:00–12:00 (KE-4-008)
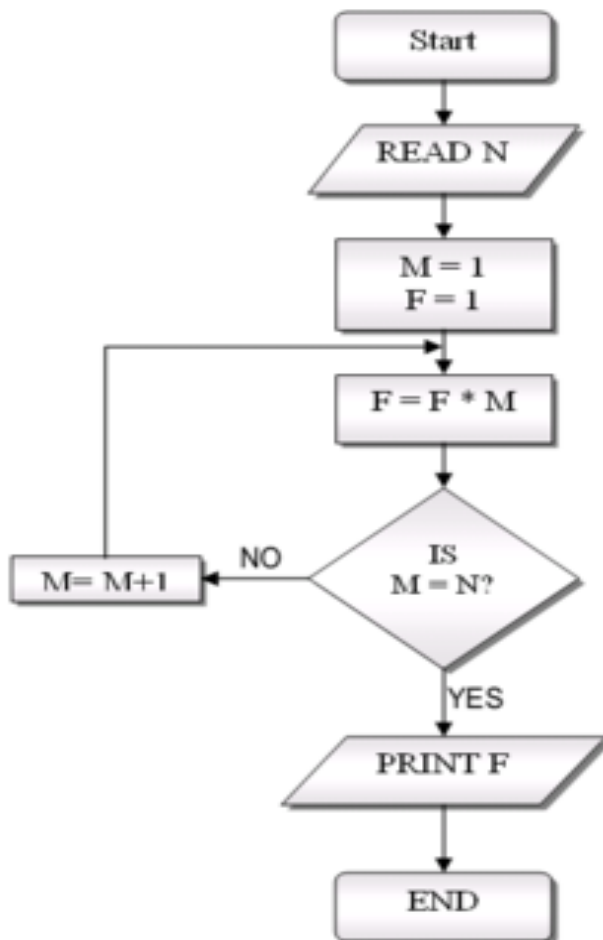
Module Timetable – Cont.

- Lab (1 hour)
  - Tuesday 11:00–12:00 (KA-1-016), Group A, Denis Manley
  - Tuesday 11:00–12:00 (Aungier St-1-005), Group B, Zanifa O Mary
  - Tuesday 11:00–12:00 (KA-3-005), Group C, Qi Wang
  - Tuesday 11:00–12:00 (KA-1-017), Group D, Cindy Liu
  - Tuesday 11:00–12:00 (Aungier St-1-006), Group E, Pierpaolo Dondio

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 flowchat and Pseudocode

```
Start
  │
  ▼
READ N
  │
  ▼
M = 1
F = 1
  │
  ▼
F = F * M  ◄──────┐
  │               │
  ▼          NO   │
IS      ──────► M = M+1
M = N?
  │
  │ YES
  ▼
PRINT F
  │
  ▼
END
```

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%
  - Mid module assessment released – 30%

Start Getting Organised
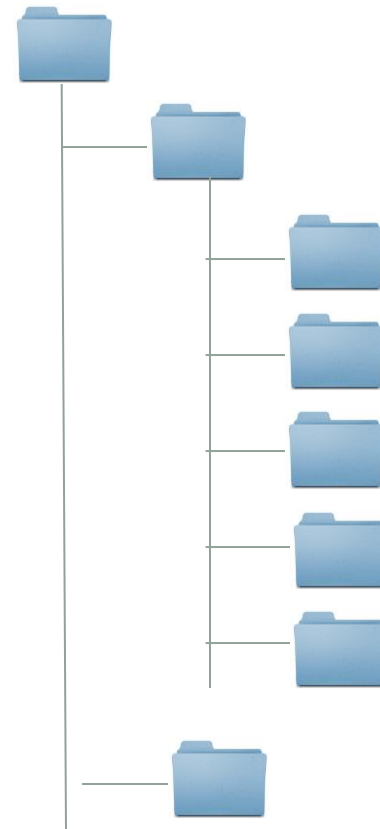
Reading Material

- 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/