Program Design

Week 1 – Module overview

Enrol in the Module on Brightspace

- All course content will be available through Brightspace.
- Self enrol on webcourses: https://brightspace.tudublin.ie/
- **Codes for enrolling on Brightspace:**
  - CMPU1024 – read the description – it should say which you should join (TU856/TU858, TU857)
  - If you cannot access the content please email Seán (me) asap!! sean.oleary@tudublin.ie

Who Is This Seán O'Leary Character?

- Undergrad DIT
- Masters University of Limerick
- PhD University of Limerick
- Previous Positions
  - Orange Labs, Brittany France
  - IRCAM/Sorbonne, Paris France
  - Maynooth University
- Lecturer Technological University Dublin School of Computer Science

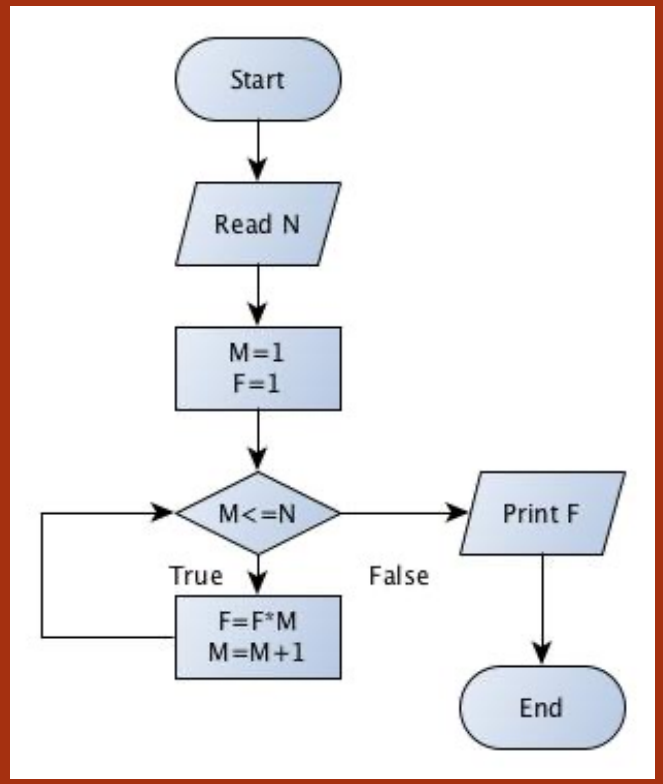
MODULE DESCRIPTION

This module is concerned with

- Program design skills
  - Flowcharts
  - Pseudocode
  - 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.

EXAMPLE OF FLOWCHART AND PSEUDOCODE

Flowchart



Pseudocode

```
Program
   total = 0

   grade_counter = 1

   While grade counter  <= 10
      Input next_grade
      total = total + next_grade
      grade_counter  = grade_counter +1
   End While
   Class_average = total /10

   Print class_average
End Program
```

MODULE AIMS

Introduction to:

- Program design techniques
- Design strategies
  - top-down
  - stepwise refinement.
- The use of pseudocode and flowcharts in program design.
- Convey an understanding and appreciation of the power of abstraction.
- The importance of a well conceived design before rushing into code

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
- Linear Data Structures -arrays.
- Boolean Algebra.
- Factorial, Greatest Common Divisor, sorting.
- Pseudocode and Flowcharts in Program Design.

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%

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;
- Problem solving and Programming Concepts; Sprankle and Hubbard

A Note on Terminology

- Generalisation
  - Making a solution applicable not just in a concrete case, but general
- Abstraction
  - Hiding the details of a solution, only making available information important in the current context
- Low/high level
  - Low level – closer to machine level
  - High level – closer to human level

General Problem-Solving Methodology

- **Define the problem**
  - Understanding the problem
  - Look at it from different angles – what data or knowledge do we have and how does this relate to our goal?
  - Restate the problem
- **Outline the solution**
  - Divide the problem – if the problem is complex can the solution be divided into smaller problems?
  - Think of analogies - have you seen something similar before?
  - How would the solution look for a simple case
  - Experiment
- **Develop the outline into an algorithm**
  - Translate the solution into computer language constructs – a form that can be put into code
  - Check each step
- **Test the algorithm for correctness**
  - Check the logic of your solution
  - Use example data to check that your solution is correct.

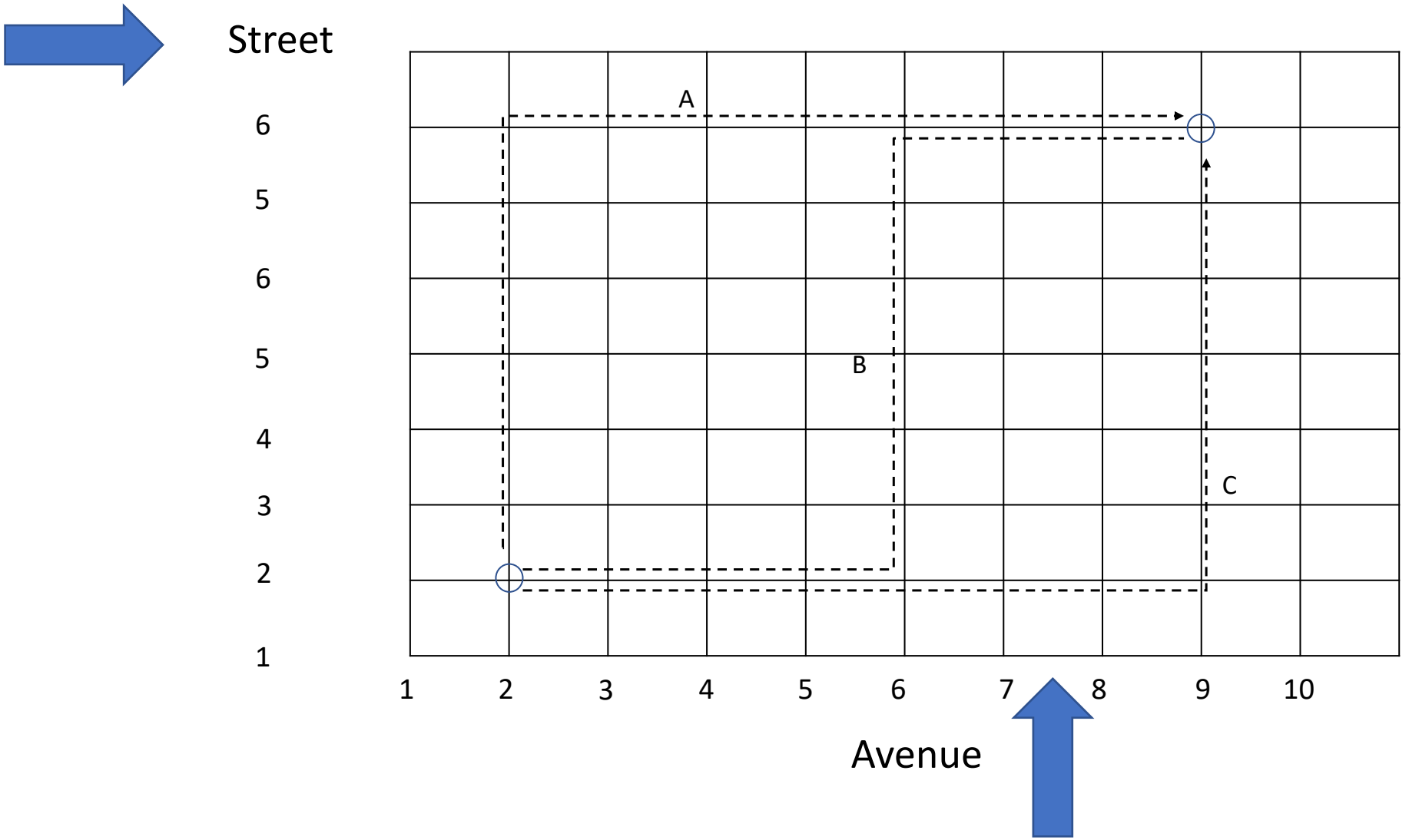
Problem solving

- A procedure of determining a solution to a problem and stating that solution in a specific programming language
- There may be numerous different solutions to a given problem.
- Analysis of a problem
  - What information is available to us regarding the problem?
  - What is the required result?
  - How can we use the initial information to achieve the required result.
- The processing is an ordered sets of steps

Example Problem – Moving from a given location to a new location on a grid.

Input	Processing	Desired Output
Location at 2nd Steet and 2nd Avenue	Change location	Location at 6th street and 9th Avenue

What We Want to Do



Biggest Number

- Which is the biggest number from these 3 values?

- 17, 64, 18

First Attempt at a Solution (C)

Input	Processing	Output
Location at 2nd Steet and 2nd Avenue	1. Go along 2nd street to 9th Avenue 2. Turn on to 9th Avenue 3. Go along 9th Avenue to 6th Street	Location at 6th street and 9th Avenue

Solution A – a more robust solution

Input	Processing	Output
Location at 2nd Steet and 2nd Avenue	1. Go from 2nd street to the next street. 2. If the next street is 1st street turn around. 3. Walk forward to 6th street 4. Turn right on 6th street. 5. If the next avenue is 1st Avenue turn around 6. Walk forward onto 9th Avenue. 7. Stop	Location at 6th street and 9th Avenue

An Alternative Solution (B)

Input	Processing	Output
Location at 2nd Steet and 2nd Avenue	1. Go from 2nd Avenue to the next Avenue on 6th street. 2. If the next Avenue is 1st Avenue turn around. 3. Walk forward to 6th Avenue. 4. Turn left on 2nd street at 6th Avenue. 5. If the next street is 1st Street turn around. 6. Walk forward onto 6th Street. 7. Turn left on 6th Avenue at 6th street. 8. If the next street is 5th Street turn around. 9. Walk forward on 6th Street to 9th Avenue. 10. Stop	Location at 6th street and 9th Avenue

- If we know enough about the problem can we make simplifications in some possible solutions?
- In solution A and B we could have learnt the direction of the streets and avenues from the first attempts and remembered them for the rest of the solution.