

# PROGRAMMING

CT103

Dr. Karl Mason

[Karl.Mason@universityofgalway.ie](mailto:Karl.Mason@universityofgalway.ie)

---

# How to contact me

- Dr. Karl Mason
  - [karl.mason@universityofgalway.ie](mailto:karl.mason@universityofgalway.ie)
  - Room 418 Top Floor IT Building

# Lecturer Bio



- Lecturer: Dr. Karl Mason
  - Currently: Lecturer Above the Bar at University of Galway
  - Prior to this role, worked at:
    - Cardiff University, UK.
    - Georgia Institute of Technology, USA.
- Research:
  - Algorithms: Evolutionary computing, Neural networks, Reinforcement learning, etc.
  - Applications: Renewable energy, Robotics, Infrastructure planning, Cloud computing, etc.
- Links:
  - University of Galway: <https://www.universityofgalway.ie/our-research/people/computer-science/karlmason/>
  - Research: <https://scholar.google.com/citations?user=kxcJG50AAAAJ&hl=en>
  - Personal: <https://karlmasonsite.wordpress.com/>

# Course Info

- Lectures – 2 hours per week
  - **Monday 11 am**, Dillon Theatre.
  - **Monday 4pm**, IT125G.
- Labs – 2 hours per week
  - **Tuesdays**
  - **Group 1: 2pm-4pm (IT106)**                      **Group 2: 4pm-6pm (IT102)**
  - **Starts 26<sup>th</sup> September**, none in Week 1.
  - Group 1 and 2 information in the coming slides.

# Canvas

- All lecture material will be uploaded to Canvas.
- This is an **online learning platform**.
- Make sure you can access Canvas!
  - You need it to access assignments.
  - If you can't, you should contact your year coordinator right away.
- I will be making announcements relating to assignments, lectures, etc. through Canvas.
  - Therefore you should **check Canvas regularly**.

# Timetable

B.Sc. in Computer Science & Information Technology

1BCT Timetable 2023/2024 Year 1, Semester I					
Time/Day	Monday	Tuesday	Wednesday	Thursday	Friday
9:00 am					
10:00 am	EE130 Fund of EEE: ENG-2001	[A] MA160 Mathematics: Tyndall Theatre [B] MA190 Mathematics: Anderson Theatre	[A] MA160 Mathematics: AC213 [B] MA190 Mathematics: Fottrell Theatre	[A] MA160 Mathematics: Dillon Theatre [B] MA190 Mathematics: Fottrell Theatre	CT102 Algor & Info. Syst.: IT125G
11:00 am	CT103 Programming: Dillon Theatre	CT101 Computing Systems: ENG G018		CT1114 Web Development: BLE2012, Block E	CT102 Algor & Info. Syst.: IT125G
12:00 pm				CT1114 Web Development: BLE2012, Block E	CT1112 Professional Skills I: AC202
1:00 pm	[A] MA160 Mathematics: IT125 [B] MA190 Mathematics: IT250	CT101 Computing Systems: Joseph Larmor Theatre	CT102 Algorithms & Information Systems Tutorial: AC214		
2:00 pm		CT103 Programming Lab: IT106	CT101 Computing Systems Lab: IT102		
3:00 pm	EE130 Fundamentals of EEE: ENG-2001	CT103 Programming Lab: IT106	CT101 Computing Systems Lab: IT102		
4:00 pm	CT103 Programming: IT125G	CT103 Programming Lab: IT102	CT101 Computing Systems Lab: IT102		
5:00 pm		CT103 Programming Lab: IT102	CT101 Computing Systems Lab: IT102		
6:00 pm					

<http://www.cs.universityofgalway.ie/timetables/>

Please note that labs will commence when advised by lecturers.

Tuesday 4 July 2023

Version 1

# Lab Info

- We will be using Microsoft Visual Studio in the labs to do the assignments
  - You will have access to these tools via the College licence (<http://nuigalway-engineering-dreamspark.onthehub.com/>)
  - There are many other tools that can also be used (basically called “C Compilers” – more on that later)

# Visual Studio

- Please install visual studio before the labs next week.
- The documents on Canvas (below) should help you do this:
  - Installing Visual Studio
  - CT103 Week 1 Exercise
- Note: If you are using a Mac, you will need to use Xcode.
  - This is explained in “CT103 Week 1 Exercise”
- This is not an assignment and you won’t be graded. However you will need to do this.



# Lab Groups

- **Group 1:**
  - 2pm to 4pm surnames **A to K**
- **Group 2:**
  - 4pm to 6pm surnames **L to Z**

# Book

- Course text:
  - Any decent introduction to programming in C will do
    - “Absolute Beginners Guide to C” by Greg Perry, Published by SAMS
    - “C for Dummies” by Dan Gookin, published by Wiley
    - “C How to Program” by Deitel & Deitel, published by Prentice Hall
- **DO NOT get a book on C++ or C# by mistake**
- **Plenty of C books in the library to borrow for free**

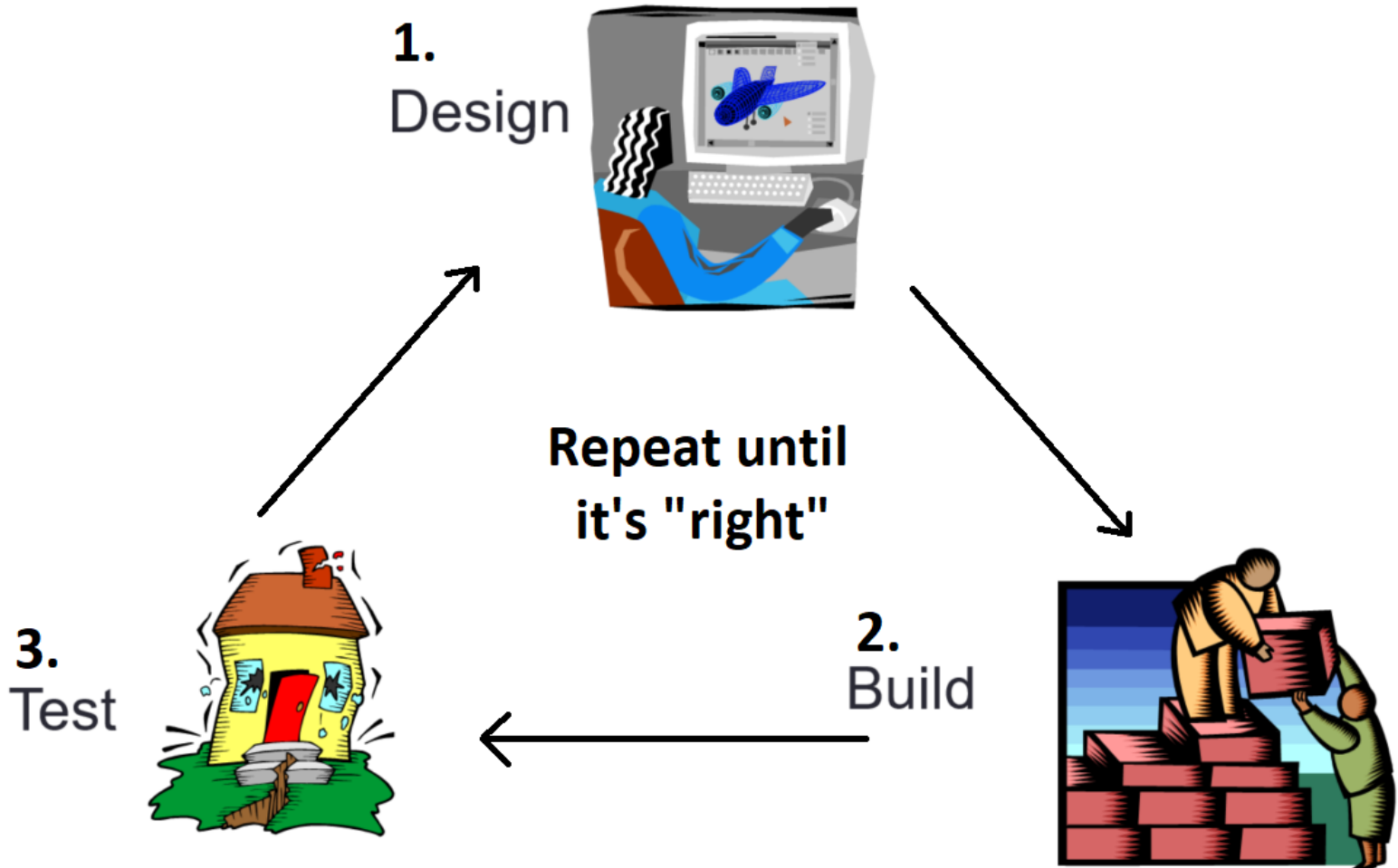
# Marking

- Lab Assignments – 25%
  - Submitted on Blackboard at the end of each lab.
  - Lab assignments in semester 1 **and** 2.
- Written exam – 75%
  - End of semester **2** only.

# Algorithm

- An Algorithm is a sequence of instructions for the computer to follow
- It usually describes:
  - The inputs you need to accomplish the task
  - The formula you need to apply to the inputs or any other manipulation of the inputs required
  - The end result or output

# Steps involved in writing software



# Programming building blocks

- Lists of instructions
  - Like cooking, e.g. “beat eggs; add flour and sugar; mix; pour into baking tin; bake at 180 for 20 minutes”
- **IF Statements** / Conditions
  - Like “IF it is raining, take an umbrella”
- **Loops** - Repeating behavior
  - 2 loop types: **For** loops and **While** loops.
  - For example “jump up and down 3 times”, or “while there is petrol left, keep driving”
- Computing results
  - Performing a sequence of steps in a particular order to get the result
  - For example to calculate your BMI divide your weight by your height squared

# PseudoCode

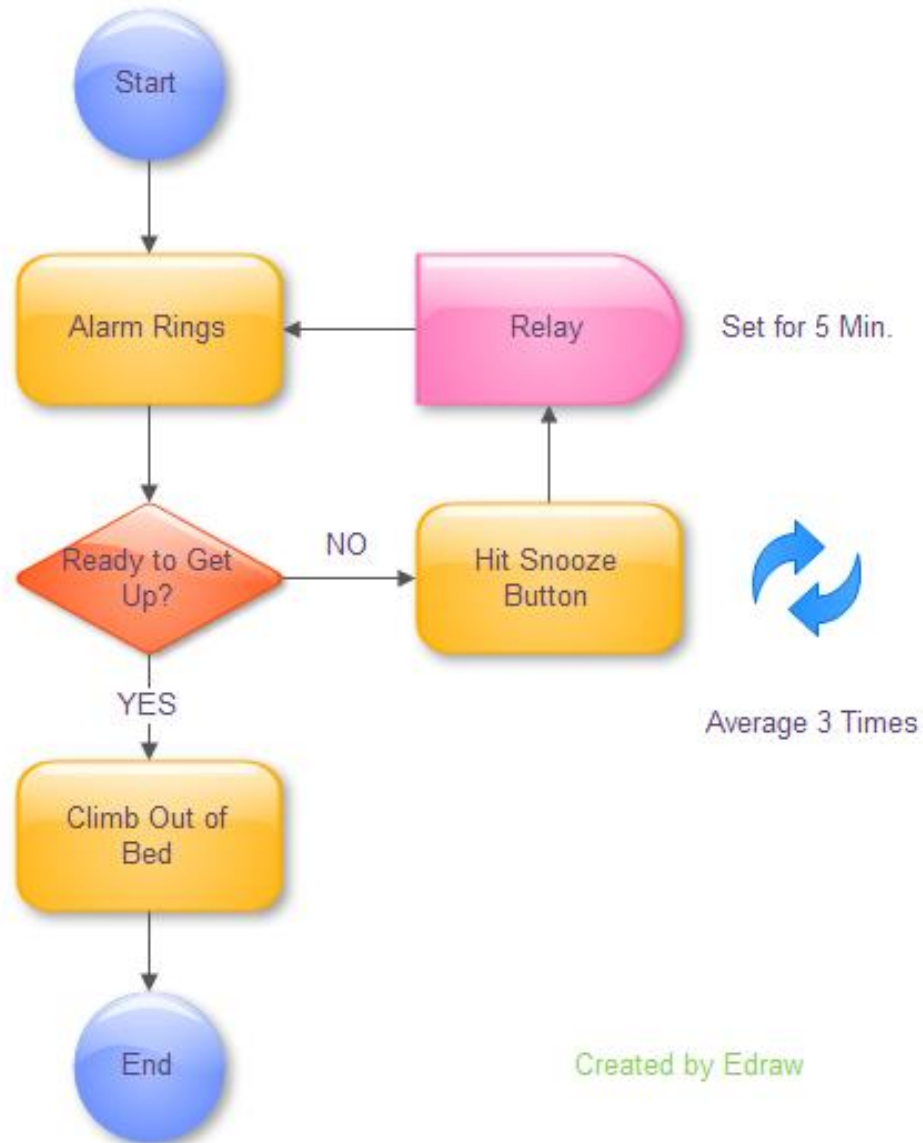
- Example #1 - Computing Sales Tax
  - Pseudo-code for task of computing the **final price** of an item after calculating **sales tax**. Note the three types of instructions: input (get), process/calculate (=) and output (display)
    1. get price of item
    2. get sales tax rate
    3. sales tax = price of item x sales tax rate
    4. final price = price of item + sales tax
    5. display final price
    6. halt
  - Variables: price of item, sales tax rate, sales tax, final price
  - Note that the operations are numbered and each operation is unambiguous
  - We also extract and list all variables used in our pseudo-code. This will be useful when translating pseudo-code into a programming language

# PseudoCode

- Example #2 - Computing Weekly Wages:
- **Gross pay** depends on the pay rate and the number of hours worked per week. However, if you work more than 40 hours, you get paid time-and-a-half for all hours worked over 40. Pseudo-code the task of computing gross pay given pay rate and hours worked.
  1. get hours worked
  2. get pay rate
  - 3 if hours worked  $\leq$  40 then
    - 3.1 gross pay = pay rate x hours worked
  - 4.else
    - 4.1 gross pay = pay rate x 40 + (1.5 x pay rate x (hours worked - 40))
  - 5.display gross pay
  - 6.halt
- Variables: hours worked, pay rate, gross pay



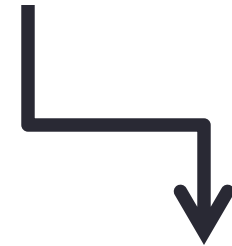
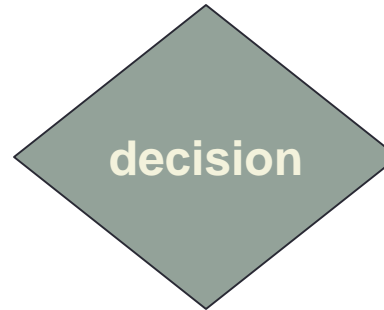
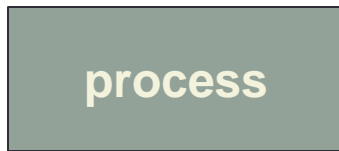
# Flowcharts



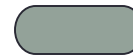
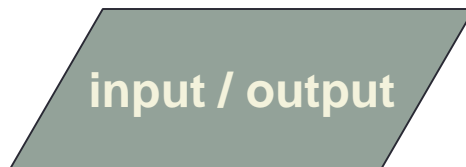
[www.edrawsoft.com](http://www.edrawsoft.com)

# Planning a program - flowcharts

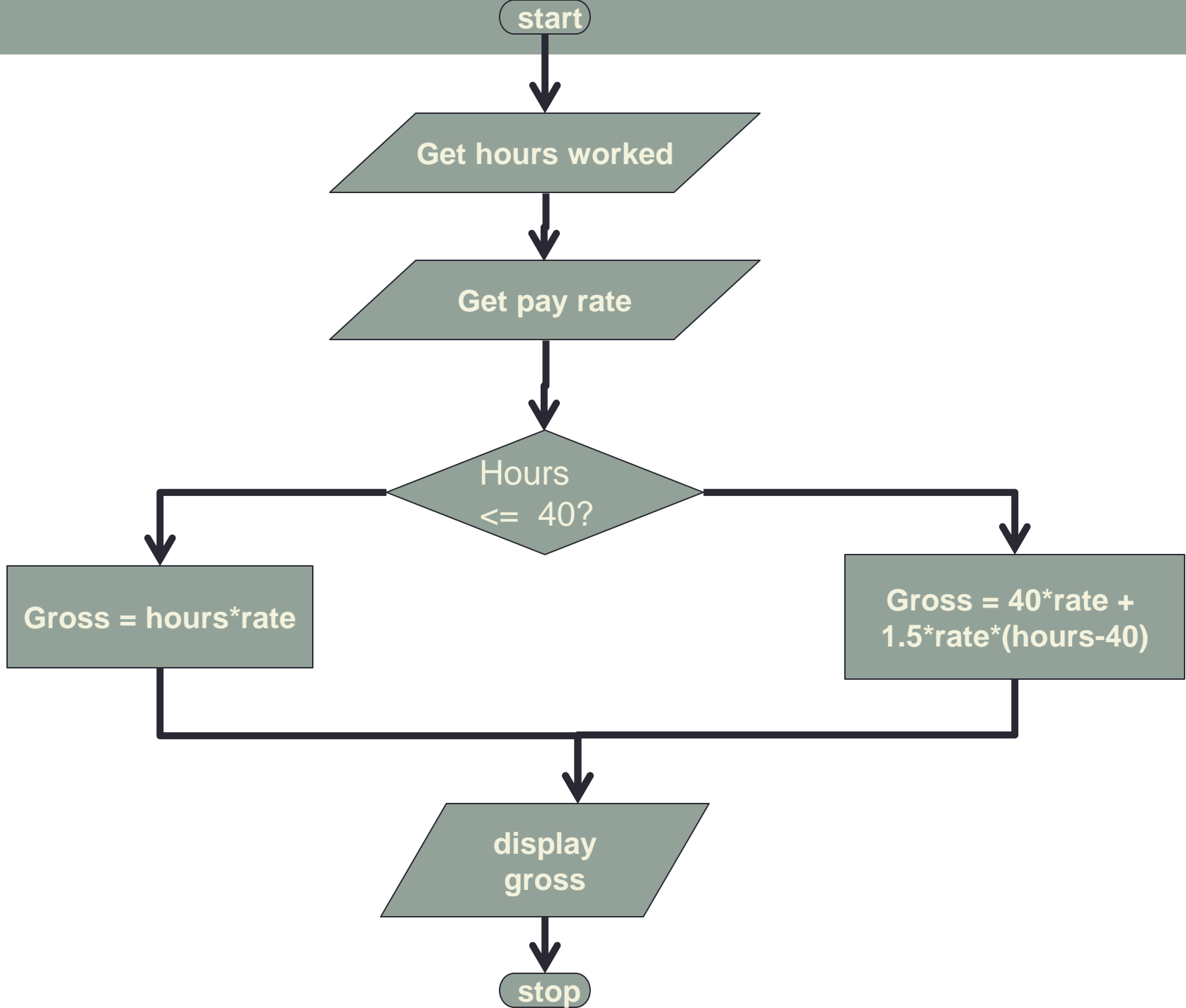
- Popular symbols:



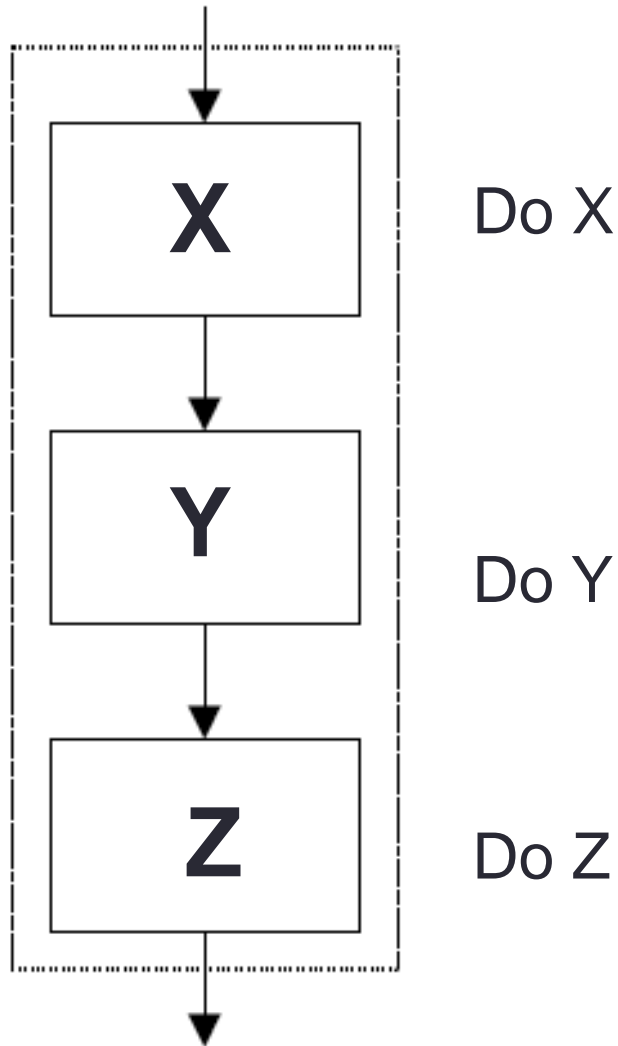
flow line



**Terminator**  
**(start / stop)**

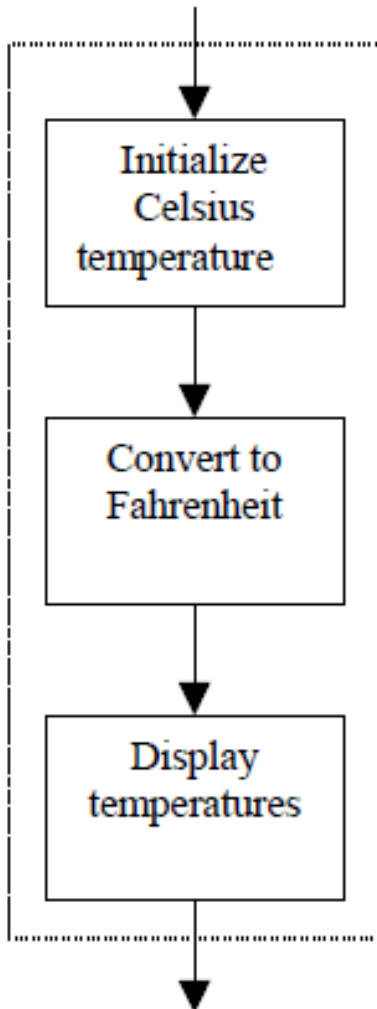


# Sequence



# Sequence Flowchart to C Code

## Flowchart



## Pseudocode

- Set Celsius temperature to 10
- Convert to Fahrenheit temperature
- Display temperatures

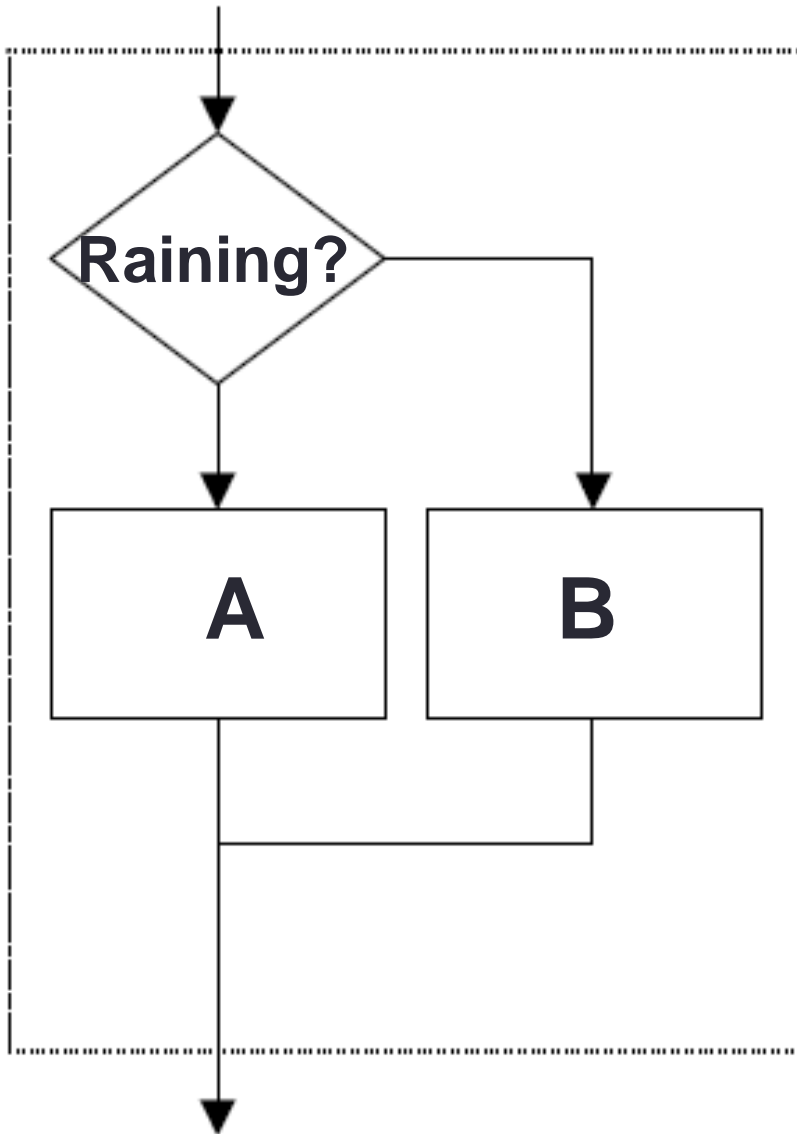
## C Code

```
x = 10;
```

```
y = 9*x/5+32;
```

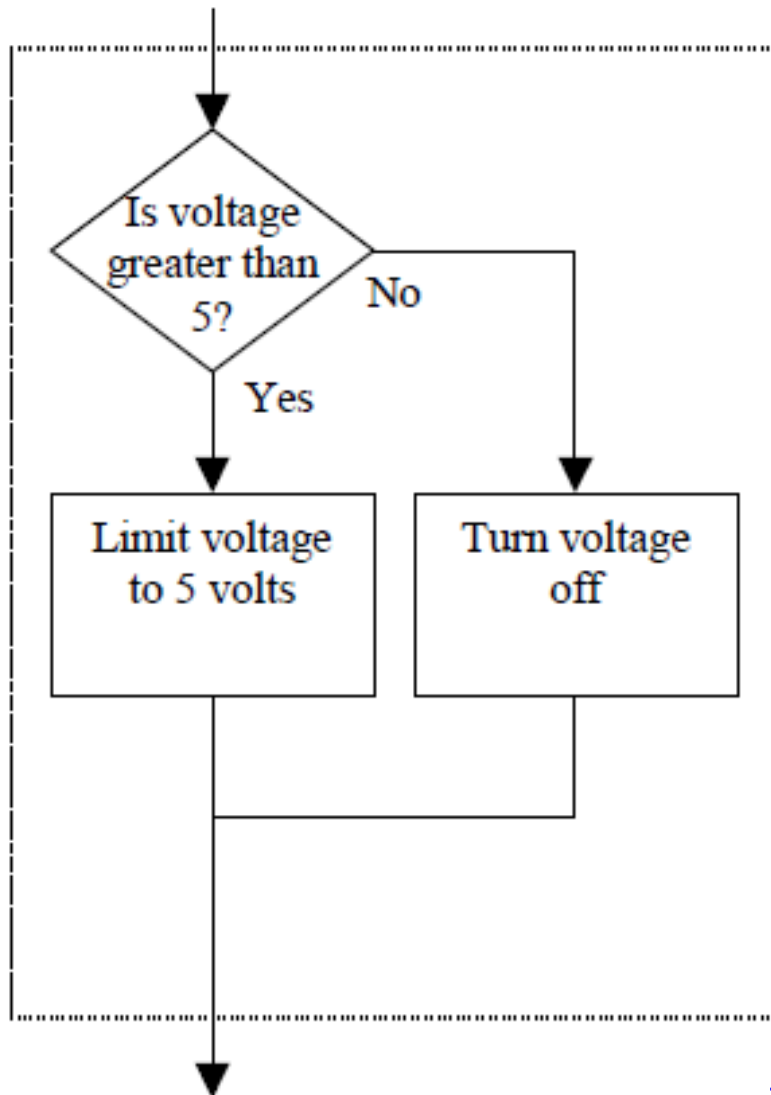
```
DispTemp(x, y);
```

# IF-THEN-ELSE



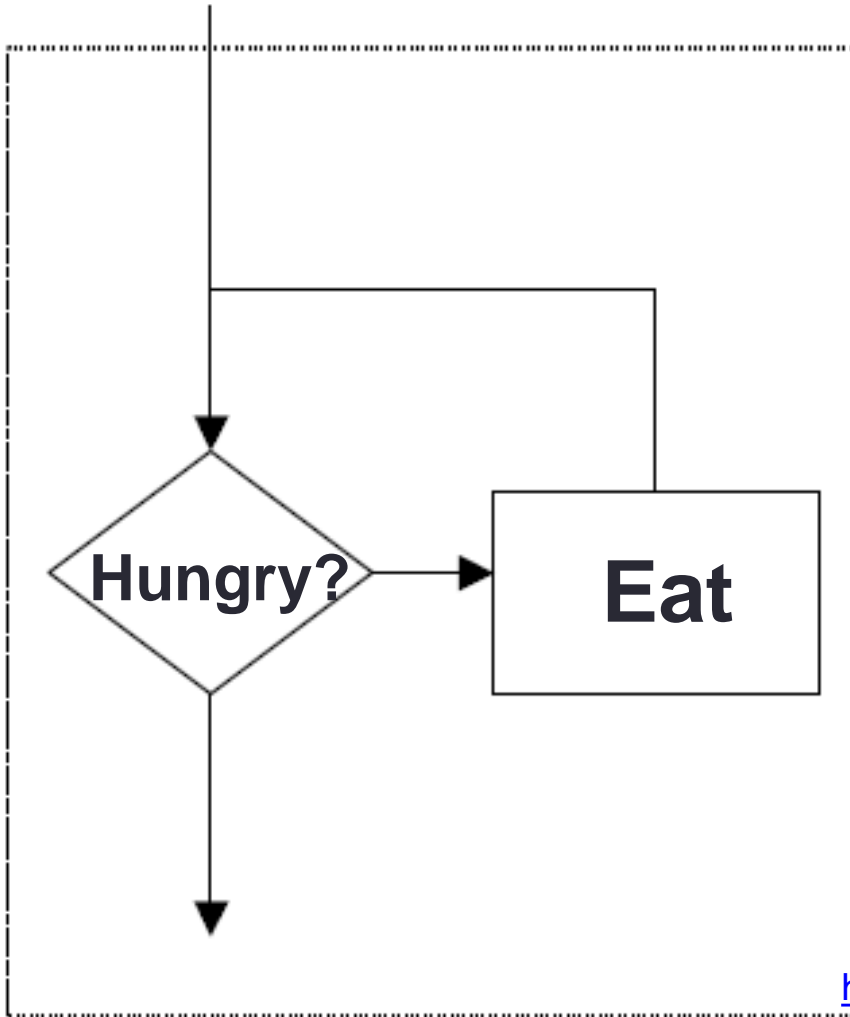
- IF Raining? TRUE THEN
  - Do A
- Otherwise
  - Do B

# IF Statement Example



```
if ( x > 5 )  
{  
    x = 5;  
}  
else  
{  
    x = 0;  
}
```

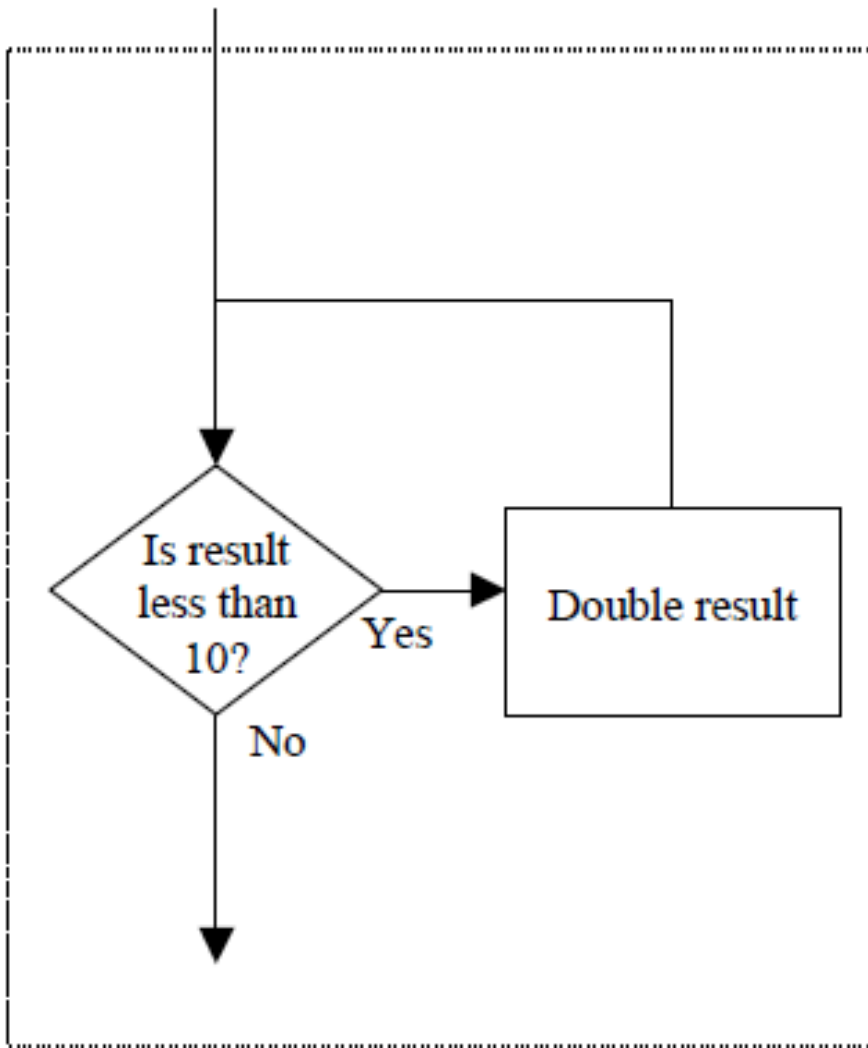
# WHILE



- While Hungry
  - Eat

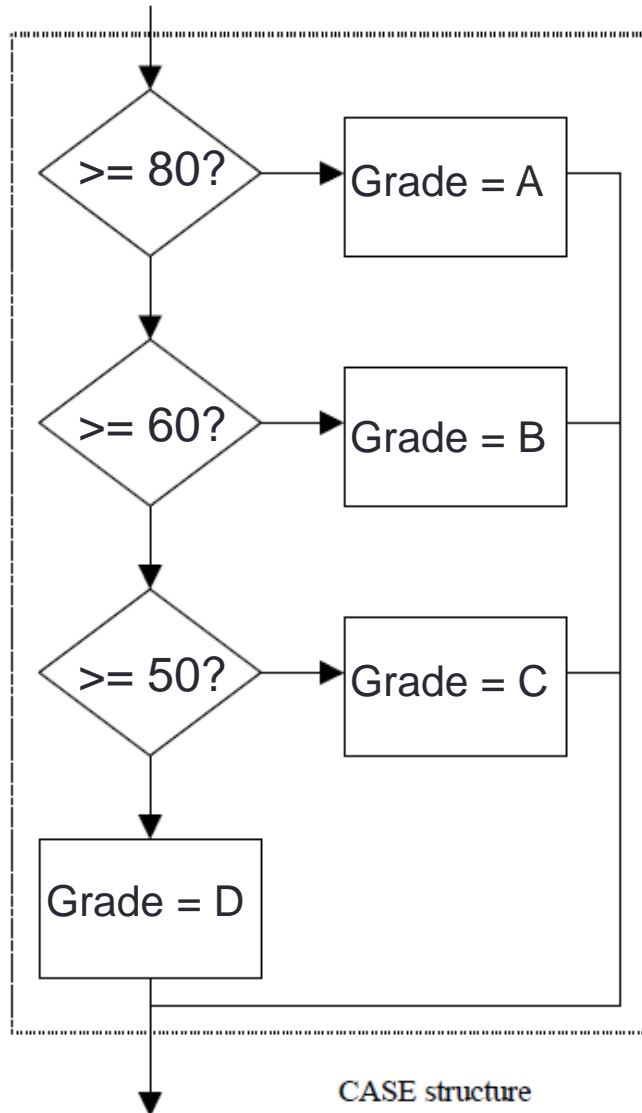


# example



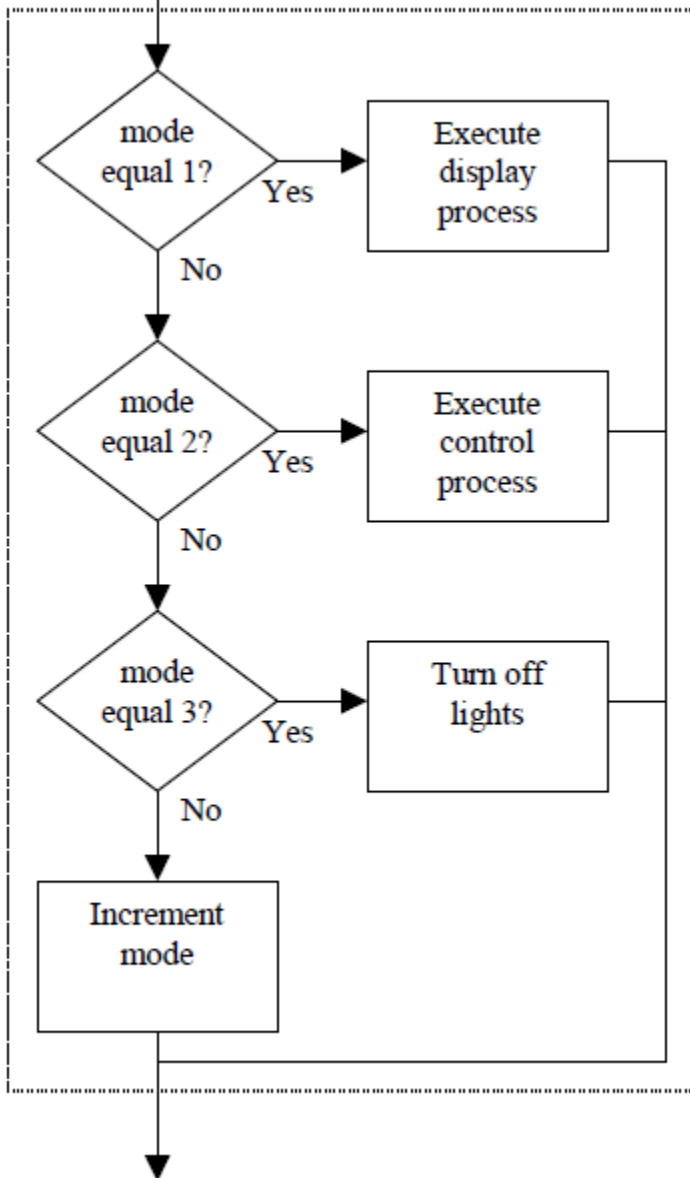
```
while ( x < 10 )  
{  
    x = 2*x;  
}
```

# CASE



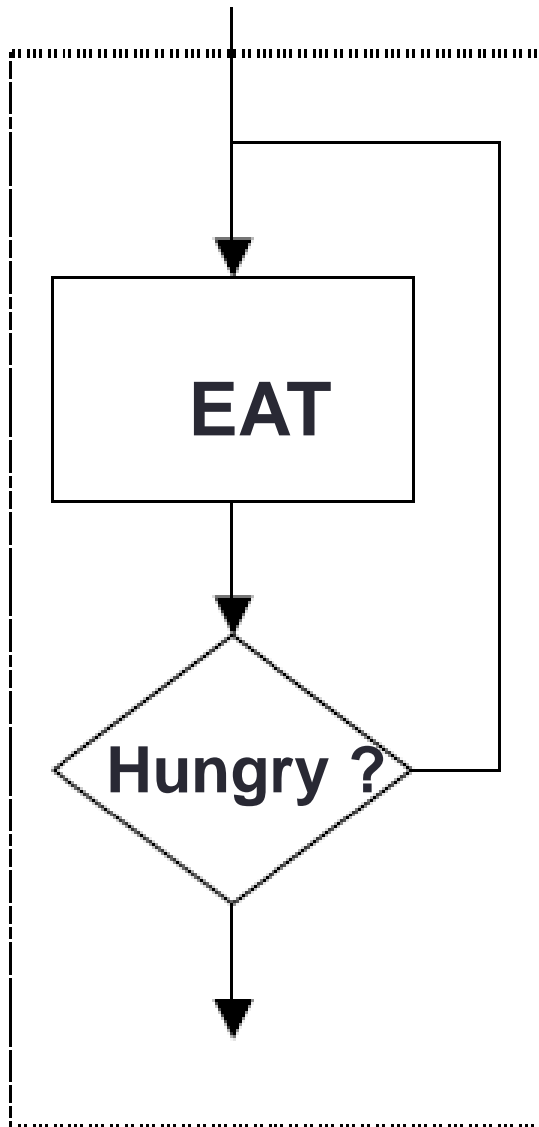
**IF** Mark  $\geq$  80? **THEN** Grade = A  
**ELSE**  
**IF** Mark  $\geq$  60? **THEN** Grade = B  
**ELSE**  
**IF** Mark  $\geq$  50? **THEN** Grade = C  
**ELSE**  
Grade = D  
**END IF**

# CASE



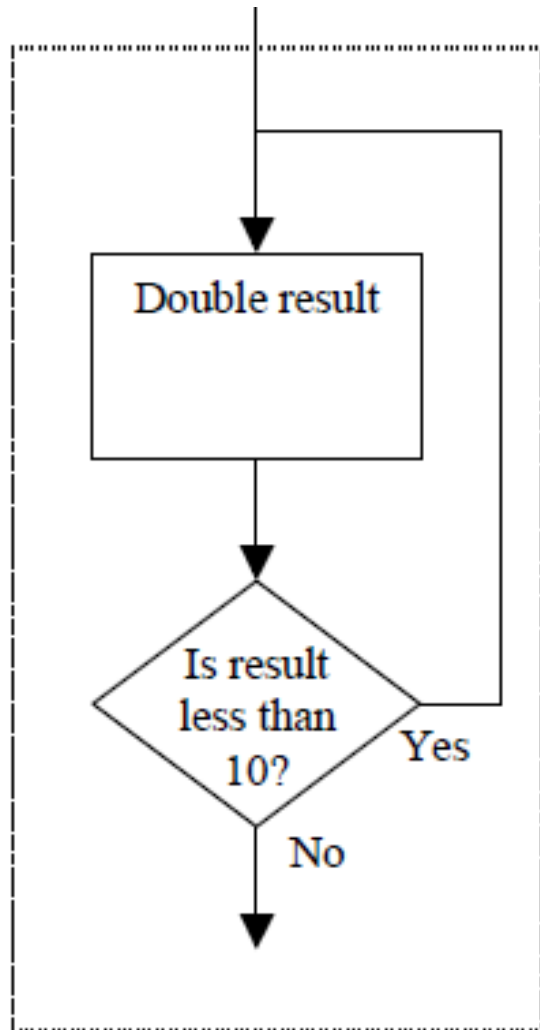
```
switch (mode) {  
  case 1:  
    /* Display process */  
    break;  
  case 2:  
    /* Control process */  
    break;  
  case 3:  
    red_light = 0;  
    blue_light = 0;  
    break;  
  default:  
    mode = mode + 1;  
    break;  
}
```

# DO-WHILE



- **Do**
  - EAT
- **While** Hungry = TRUE

# DO-WHILE



```
do
{
    x = 2*x;
}
while ( x < 10 );
```

# Algorithm to calculate factorial of a number

1. Start
2. Read the number  $n$
3. [Initialize]  
     $i=1$ ,  $fact=1$
4.  $i=i+1$
5.  $fact=fact*i$
6. Repeat step 4 through 6 until  $i=n$
7. Print  $fact$
8. Stop

# Flowchart

