Intro to Programming

C/C++

Sankalp Gupta

moklaeducation@gmail.com

Learning Goals

- Comfort with
  - Basics of C/C++
  - Basics of Computer Science
- Familiarity with IDE
  - Visual Studio Community Edition
- Thinking logically
  - i.e. One step at a time
  - And Visualizing how computer works
- Independence of technology

Why C/C++

- Foundational understanding
  - Understand Computer Science
- Speed and control
  - Fastest programming language
- Really small programming language
  - C has 32 keywords
  - C++ has 92 keywords as of 2023
- Makes you digital native

When NOT C/C++?

- Slower manual speed of writing code

- Don't care about speed

- Don't care about deep Computer Science
  - Although this may not be achievable

Bit : Smallest unit of memory

- Bits are used for representing everything
- Have 2 states : 0 and 1 , like a bulb
  - On : 1
  - Off : 0
- Nibble : 4 bits
- Byte : 8 bits
- int (integer) 4 bytes
- char (character) 1 byte

Hertz : Unit of time and speed in Computers

- 1 Hertz : once per second
  - 1 unit of work per clock instruction
- Modern processors
  - Measured in Giga hertz
  - High Core Counts
  - More instructions

Hello World

- Let's go ,
- Program to print "Hello World"
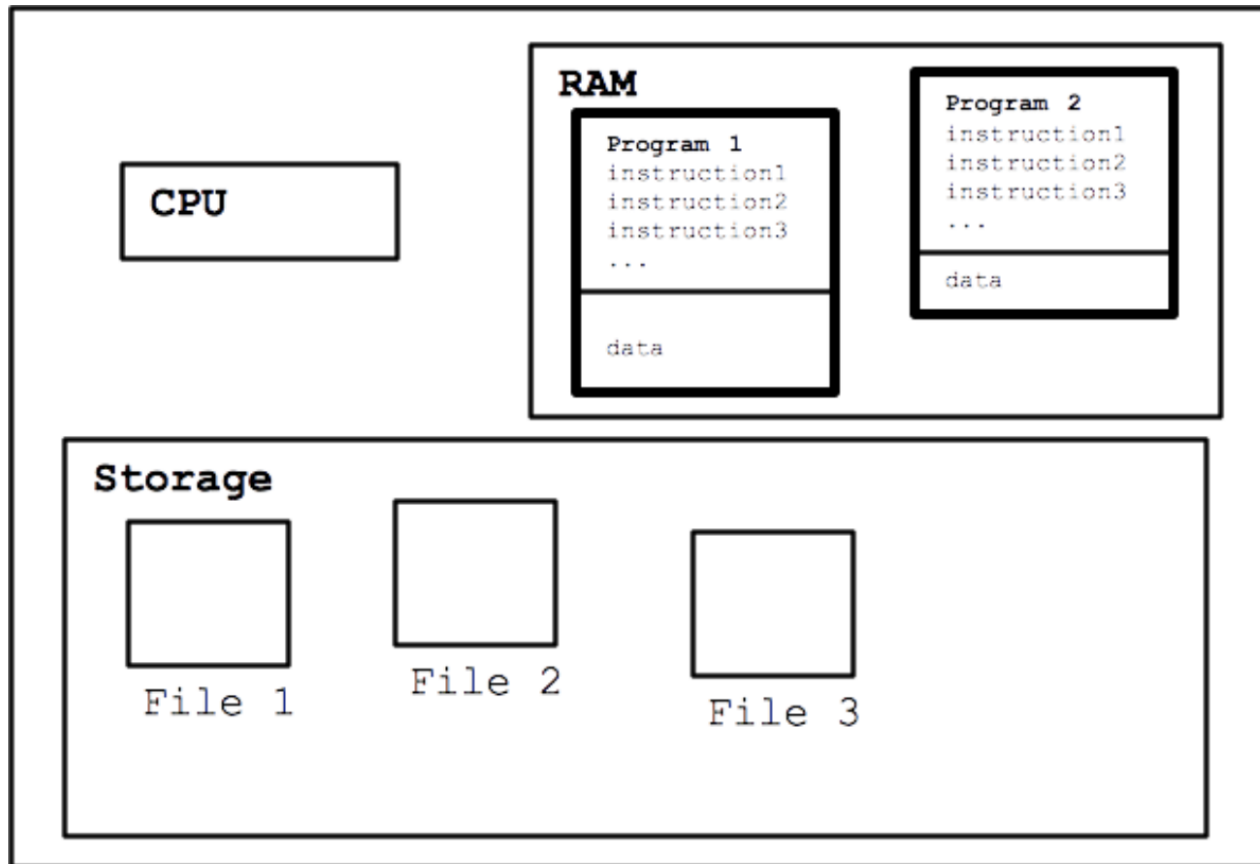

#include <iostream>

using namespace std;

int main()
{
    cout << "Hello World";
    return 0;
}

Print a diamond pattern

```
   *
  ***
 *****
  ***
   *
```

Running the program : model

Data Types(Primitive/built in)

- bool
- int
  - short(2), long(4)
  - signed,unsigned
- float
  - float(4), double(8) , long double(8,10,16)
- char(1)
  - signed,unsigned
  - wchar_t(2)
- void

Variables

- Containers for storing data
- Value can change during execution (unless you don't want it to)
- Declaration
  - int birthyear;
  - float weight;
  - char courseGrade;

Data Types (Derived)

- Arrays
  - char name[100];
    - Size 100
    - Index : 0 - 99
  - int age[10];
    - Size 10
    - Index : 0 - 9
  - float power[20]
    - Size 20
    - Index : 0 - 19

Integer vs floating math

- Division
  - Float :continuos, contains decimal point
  - Int : discrete , truncates everything after decimal ,
- float f = 10;
  - cout << f/3;
  - 3.33
- int i = 10;
  - cout<< i/3;
  - 3

Type Name	Bytes	Other Names	Range of Values
int	4	signed	-2,147,483,648 to 2,147,483,647
unsigned int	4	unsigned	0 to 4,294,967,295
bool	1	none	false or true
char	1	none	-128 to 127 by default
signed char	1	none	-128 to 127
unsigned char	1	none	0 to 255
short	2	short int, signed short int	-32,768 to 32,767
unsigned short	2	unsigned short int	0 to 65,535
long	4	long int, signed long int	-2,147,483,648 to 2,147,483,647
unsigned long	4	unsigned long int	0 to 4,294,967,295
long long	8	None	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
unsigned long long	8	none	0 to 18,446,744,073,709,551,615
enum	varies	None	
float	4	None	3.4E +/- 38 (seven digits)
double	8	None	1.7E +/- 308 (fifteen digits)
long double	8	None	Same as double
wchar_t	2		0 to 65,535

User Defined Data Type

- Enum
  - Short for enumeration
  - Define a set of named, integer constants

```
enum ESpeed
{
    low,
    medium,
    high
};

ESpeed fanSpeed;
fanSpeed = low;
```

Code : Greeting with name/age

- Given user
  - Store name in char array (char name[100])
  - Store year of birth in int(int birthyear)

- Calculate age
  - age = currentYear – birthyear;

- Print Name, age.
  - cout <<"Hello "
    - << name
    - <<" I know your age is :"
    - <<age;

Advanced : Count digits in a number

- Define a number
- Return the number of digits, (assume positive)
  - 129 → 3
  - 34   → 2
  - 0   → 1
- Hints
  - You need to know loops
    - while
  - You need to know integer maths
    - division

Input , Output and Processing for Humans

- Speech
  - Process →Speak →Listen

- Book
  - Read →Process → Memorize

- Conversation
  - Listen → Speak →Listen →Speak ….

I/O and processing for Computers

- Video games
  - Input (controllers) →Process →Output (Screen)
- Movies
  - Input (network) →Process → Output (visuals, audio)
- Console
  - Input(char, int, float) → Process →Output (char(s), int, float)

Standard Input

- cin
  - Read data from keyboard
  - Store it in variables

- Extraction operator
  - >>

- Can use multiple data types (char, int, float , …)

- Example
  - int age;
  - cin >> age;

Standard Output

- cout
  - Write data to console/Screen
  - Reads from memory

- Insertion operator
  - <<

- Can use multiple data types(variables, literals, constants)

- Example
  - int age = 172;
  - cout << age;

Operators >>,<<

- Extraction and Insertion operators
- Can be cascaded
  - cin >>age >> name;
  - cout<<age << name;
- << works with stream modifiers
  - "\n" : newline
    - cout <<"\n"; //moves the cursor to new line
  - Or
    cout <<endl; //Same visual effect as "\n" but is different
  - (there are other stream modifiers too)

Lab

- Using Cin (Hint: define a variable first)
  - Input a character (char)
  - Input an integer(int)
  - Input a decimal(float)
- Using cout
  - Output a character
  - Output an integer
  - Output a decimal.
- Use endl and "\n"
- Cascade the operators

Assignment : Mad libs story

```cpp
#include <iostream> // Required for input/output operations (cout, cin)
#include <string>   // Required for using string data type

int main() {
        std::string adjective1, noun1, verb1, adjective2, noun2;

        // Prompt the user for input and store it in the variables
        std::cout << "Enter an adjective: ";
        std::cin >> adjective1;

        std::cout << "Enter a noun: ";
        std::cin >> noun1;

        std::cout << "Enter a verb: ";
        std::cin >> verb1;

        std::cout << "Enter another adjective: ";
        std::cin >> adjective2;

        std::cout << "Enter another noun: ";
        std::cin >> noun2;

        // Construct and print the Mad Libs story
        std::cout << "\n--- Your Mad Libs Story ---\n";
        std::cout << "Once upon a time, there was a " << adjective1 << " " << noun1
                << ". It loved to " << verb1 << " all day long.\n"
                << "One day, it met a " << adjective2 << " " << noun2
                << ", and they lived happily ever after.\n";

        return 0; // Indicate successful program execution
}
```

Boss Assignment

- Input student details
  - Student Name
  - Subject name
  - Marks (out of 100)
- Process
  - Find grade using this table
  - 90 <marks →A
  - 75 <= marks < 90 →B
  - 60 <= marks <=74 →C
  - marks <60 →D
- Output
  - Grade for the student
- Challenge
  - Enter multiple students, print how many students had A, B , C and D grades each.
- Hint
  - Need to know conditional (if-else)
  - May need to know loop (while)

Recall , Review

- C/C++Keywords we know already
  - And few more
    - signed , unsigned
    - short , long
- Computers think in 0s and 1s
  - What types ?
- Computer have speed measured in hertz (Hz)
  - How much faster is 1Kilo Hertz than 1Hz ?
  - What are current computer speeds ?

Measuring computer capabilities (some more units)

- FLOPS : floating point operations per second
  - FP 16, **FP 32**, FP 64
- IOPS : Input/Output operations per second
- Fun facts
  - Computers are afraid of floats
  - Computers are afraid of division

Computer trends

- Getting smaller
  - **Die** sizes have been shrinking
- Getting faster
  - Same size **die** have more **transistors**
- Getting crowded
  - More **core** counts per **die**
- Getting chatty
  - Networked , internet connected
- Getting efficient
  - More performance per unit of **power**

Back to C++, Decision Making & Branching
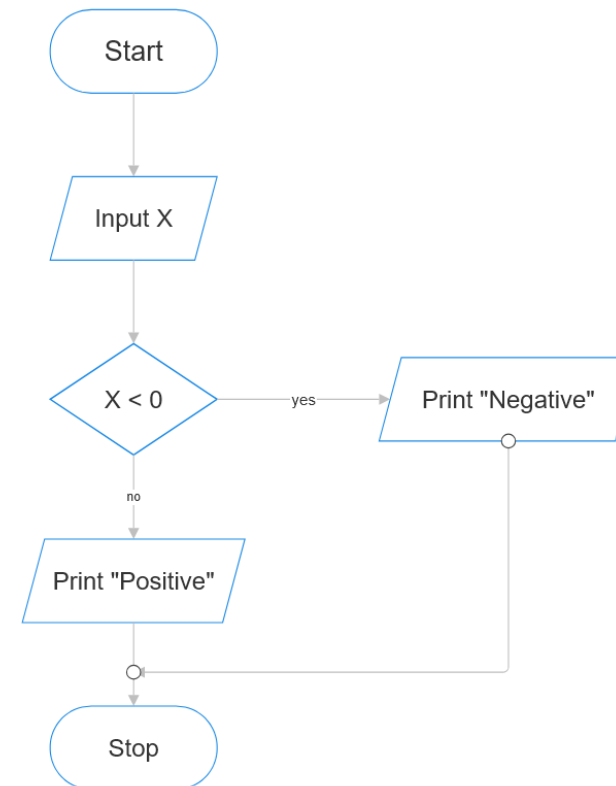
- if , else
- switch
- goto

Making Decisions , using "if"

```cpp
int x;
cout << "Enter a number :";
cin >> x;

if (x < 0)
{
    cout << "\nThis is a negative number";
}
```
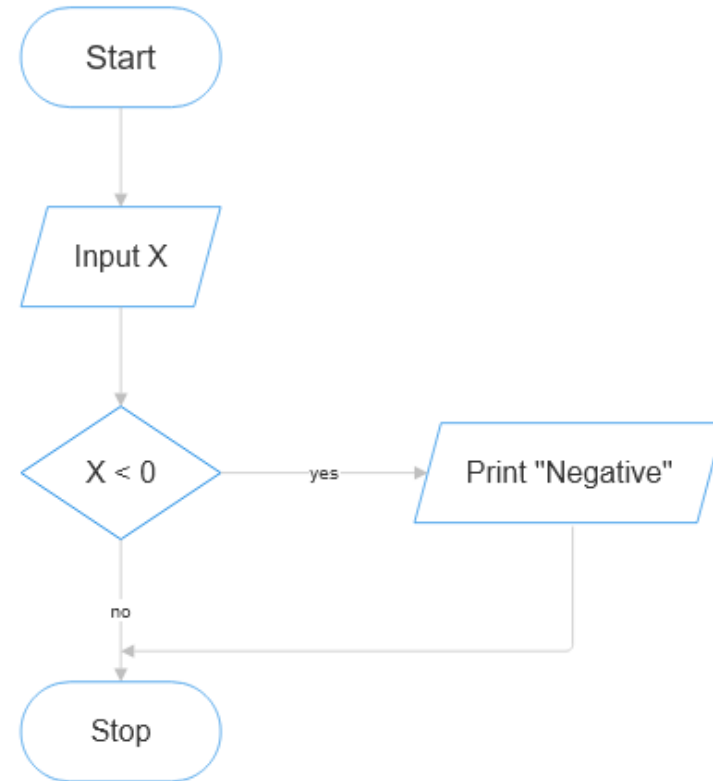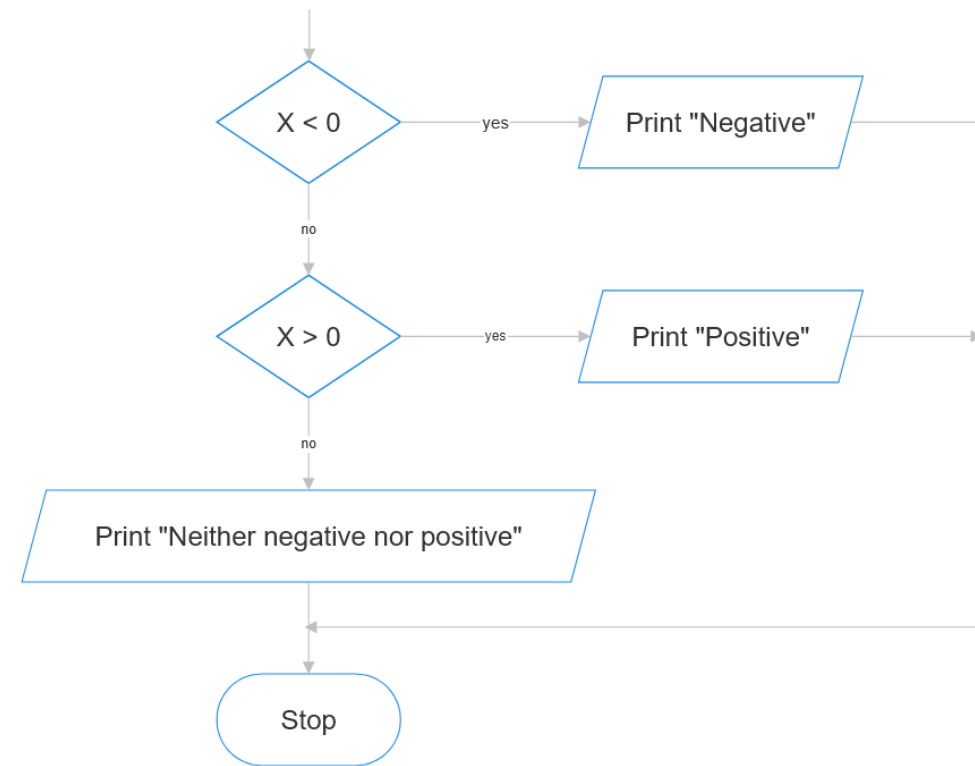
Making Decisions using "if else"

```cpp
int x;
cout << "Enter a number :";
cin >> x;

if (x < 0)
{
    cout << "\nThis is a negative number";
}
else
{
    //Bug ?
    cout << "\nThis is a positive number";
}
```

Making decision, If else if else

```cpp
int x;
cout << "Enter a number :";
cin >> x;

if (x < 0)
{
    cout << "\nThis is a negative number";
}
else if(x >0)
{
    cout << "\nThis is a positive number";
}
else
{
    cout << "\nThis is a neither negative or positive";
}
```
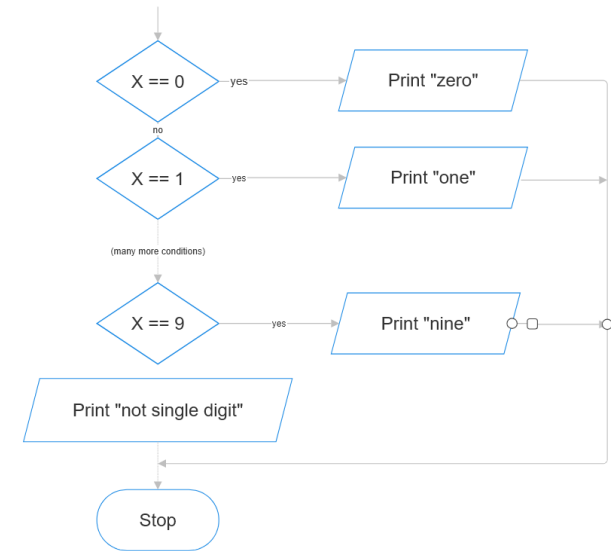
Relational operators

- less than
  - <
- greater than
  - >
- less then or equal to
  - <=
- greater than or equal to
  - >=
- is equal to
  - ==
- is not equal to
  - !=

Lab (if else)

- Get user to enter a single digit number
- Print the digit in English
  - 0 → "zero"
  - 1 → "one"
  - …
  - 8 → "eight"
  - 9 → "nine"
  - (Anything else) "→ this is not a single digit number"

Branching using Switch

```
switch (x)
{
    case 0:
        cout << "zero \n";
        break;

    case 1:
        cout << "one \n";
        break;

    case 2:
        cout << "two \n";
        break;

    case 3:
        cout << "three \n";
        break;

    case 4:
        cout << "four \n";
        break;

    case 5:
        cout << "five \n";
        break;

    case 6:
        cout << "six \n";
        break;

    case 7:
        cout << "seven \n";
        break;

    case 8:
        cout << "eight \n";
        break;

    case 9:
        cout << "nine \n";
        break;

    default:
        cout << "not a single digit number \n";
}
```
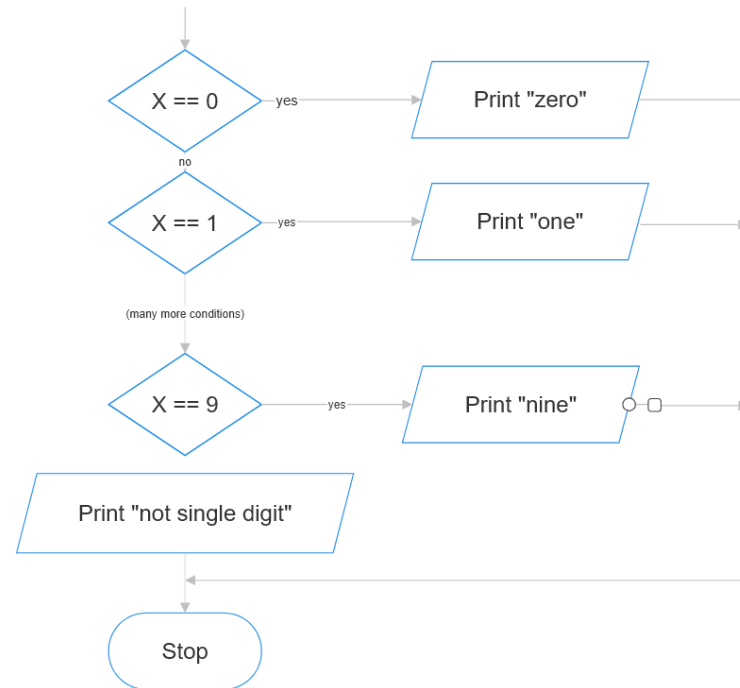
Lab (switch)

- Get user to enter a single digit number from [1,2,3,4,5,6,7]
- Convert it into day of week
  - 1 → "Sunday"
  - 2 → "Monday"
  - 3→ "Tuesday"
  - 4 → "Wednesday"
  - 5 → "Thursday"
  - 6 → "Friday"
  - 7 → "Saturday"
  - (Anything else) → "Invalid number for a day"

Asssignment , quiz game

- Create a quiz game containing atleast 3 questions.
    - Print a question ,
    - provide 4 numbered options ,
    - User enters the option number and verifies result.
    - Keep score of how many correct responses user input
- Print the score (e.g.3 /4 correct ,)
- An example
Q1. What is the capital of USA ?
    1. Seattle
    2. Los Angeles
    3. Washington DC
    4. Chicago
    Enter your response : _

Boss Assignment
Convert a string of 0,1 to decimal

- Get a string of 0s and 1s from user

- Calculate what the decimal number for it is ?

- Note
  - This is very hard , may take hours/days/weeks

- Hint
  - You would need to know
    - string (or char array)
    - Loops (while)
    - Binary logic
    - *Loops within loops

Code, as seen by computer : HelloWorld

- Cpp vs assembly

```cpp
1    #include <iostream>
2
3    int main()
4    {
5        std::cout << "Hello World!\n";
6    }
7
```

```asm
1  section .data
2      msg db 'Hello, World!', 0xA   ; Message string with newline
3      len equ $ - msg               ; Length of the string
4  section .text
5      global _start
6  _start:
7      mov eax, 4                    ; System call number for sys_write
8      mov ebx, 1                    ; File descriptor (stdout)
9      mov ecx, msg                  ; Pointer to the message
10     mov edx, len                  ; Length of the message
11     int 0x80                      ; Call kernel
12     mov eax, 1                    ; System call number for sys_exit
13     xor ebx, ebx                  ; Exit code 0
14     int 0x80                      ; Call kernel
```
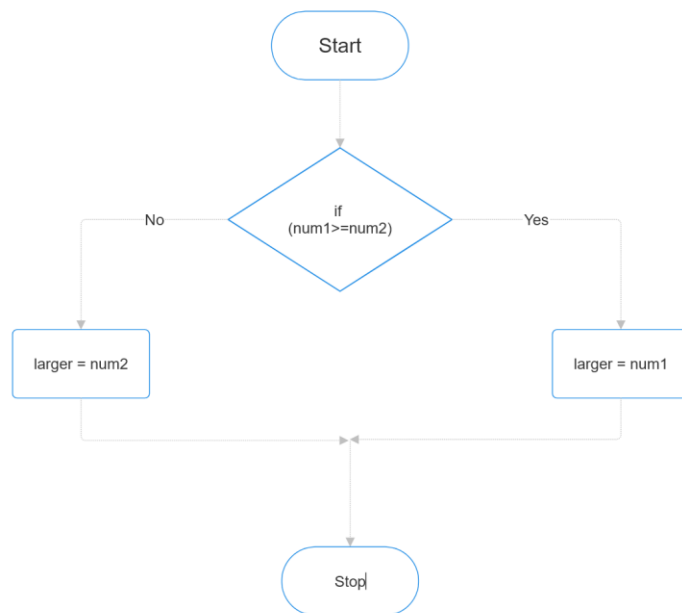
Code , as seen by computer : Find larger num.

- Cpp vs Assembly

```cpp
1   #include <iostream>
2
3   int main()
4   {
5       int num1 = 10;
6       int num2 = 20;
7       int larger = 0;
8
9       if (num1 >= num2)
10      {
11          larger = num1;
12      }
13      else
14      {
15          larger = num2;
16      }
17  }
```

```asm
1   section .data
2       num1 dd 10
3       num2 dd 20
4       larger dd 0
5   section .text
6       global _start
7   _start:
8       mov eax, [num1]     ; Load num1 into EAX
9       mov ebx, [num2]     ; Load num2 into EBX
10      cmp eax, ebx        ; Compare EAX and EBX
11      jge store_eax       ; If EAX >= EBX, jump to store_eax
12      mov eax, ebx        ; Otherwise, move EBX into EAX
13  store_eax:
14      mov [larger], eax   ; Store the larger value in memory
15      mov eax, 1          ; Exit system call
16      xor ebx, ebx        ; Exit code 0
17      int 0x80
```

Branching visualized

- Code looks sequential
- But really has branches


```cpp
1     #include <iostream>
2
3     int main()
4     {
5         int num1 = 10;
6         int num2 = 20;
7         int larger = 0;
8
9         if (num1 >= num2)
10        {
11            larger = num1;
12        }
13        else
14        {
15            larger = num2;
16        }
17    }
```

```asm
1    section .data
2        num1 dd 10
3        num2 dd 20
4        larger dd 0
5    section .text
6        global _start
7    _start:
8        mov eax, [num1]        ; Load num1 into EAX
9        mov ebx, [num2]        ; Load num2 into EBX
10       cmp eax, ebx           ; Compare EAX and EBX
11       jge store_eax          ; If EAX >= EBX, jump to store_eax
12       mov eax, ebx           ; Otherwise, move EBX into EAX
13   store_eax:
14       mov [larger], eax      ; Store the larger value in memory
15       mov eax, 1             ; Exit system call
16       xor ebx, ebx           ; Exit code 0
17       int 0x80
```
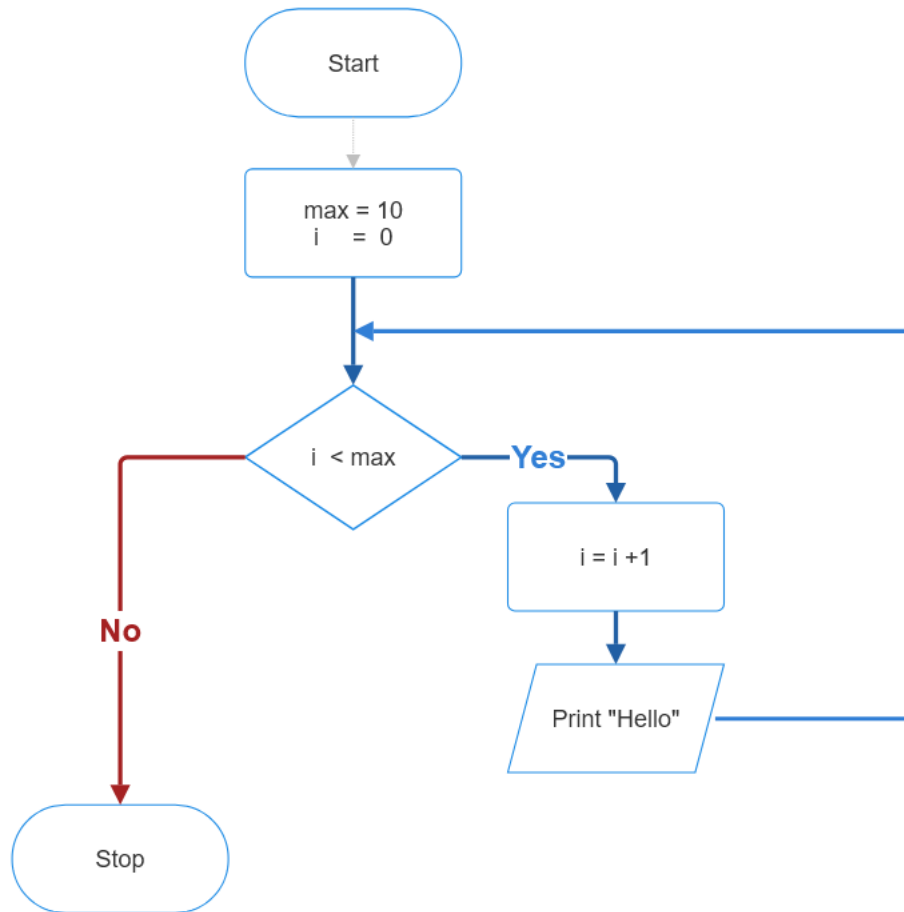
Loops : Things that happen over and over ...

- Years
  - ...,2024,2025, ...
- Seasons
  - Spring, summer, fall, winter
- School
  - Learn , Homework, Tests
  - Scoring
- Reading a book
  - Title,... page, page,page ...,the end

Decision making and looping

- Loops
  - Print "Hello"10 times
  - 10 x cout ?

- What if 100 times ?
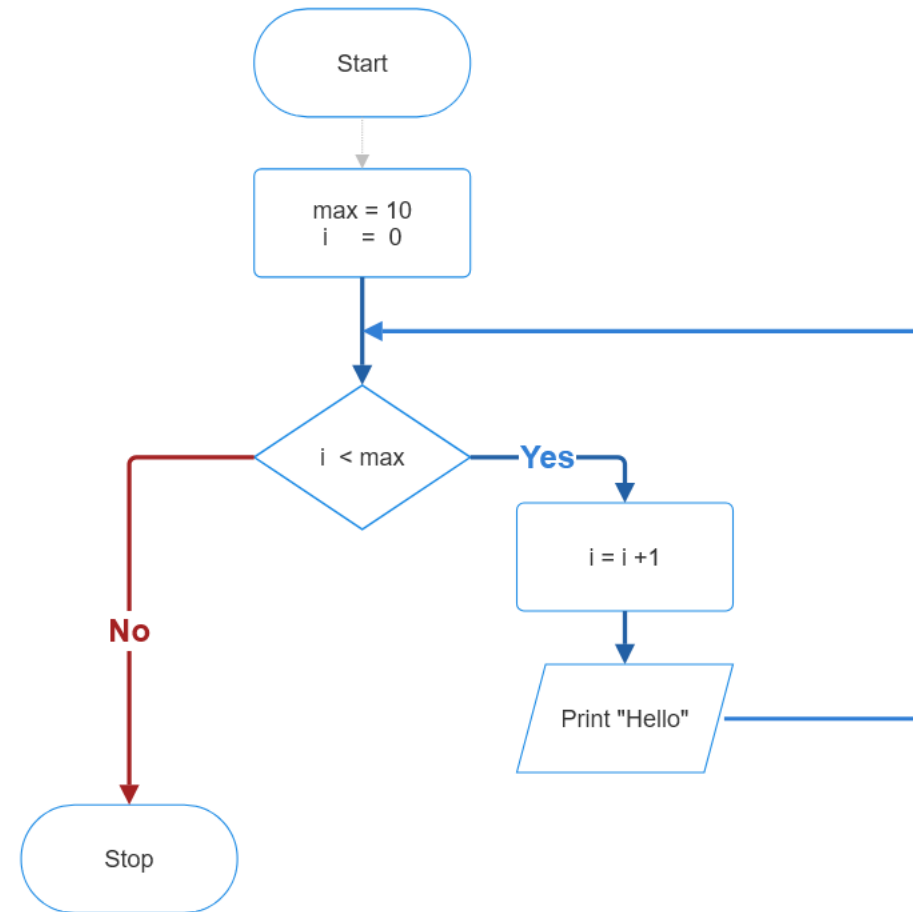 - 100 x cout ??

- Can we do better ?

Loop : visualized

- Print "Hello" 10 times

Loop : visualized

- Print "Hello" 10 times

```cpp
1    #include <iostream>
2    using namespace std;
3
4    int main()
5    {
6        int max = 10;
7        int i = 0;
8
9        while (i < max)
10       {
11           cout << "Hello " << endl;
12       }
13   }
```

Assignment 1, Guess the number

- Computer finds a random number between 0..9

- User guesses it

- If user fails .. they try again … (*use while loop)

-
```cpp
1
2    #include <iostream>
3    #include <cstdlib>
4    using namespace std;
5
6
7    int main()
8    {
9        int randomNumber = rand() % 10;
10
11       int num;
12
13       cin >> num;
14
15       //Fill in the rest ....
```

Boss Assignment

- Get a number from user
- Convert the number(int) into binary.
- Examples
  - 0 →   0
  - 1 →   1
  - 7 → 111
  - 8 →1000
  - 12→1100
  - 15 →1111

Assignment 2.* *(hint : use while)*

- Print first 10 numbers

- Print the sum of first N numbers
  - N is provided by the user

- Find the sum of first N odd numbers

- Find smallest number divisible by all numbers between 1 to 20.

- Find the greatest common factor of 2 numbers .
  - 2,4 → 2
  - 6,9 → 3

- EASYMEDIUMHARD

Decision Making at a glance

- Branching
  - If/else
    - slides 30 - 32
  - switch
  - goto
- Looping
  - while
    - slide 45
  - do/while
  - for
- Jumps in loops
  - break
  - continue

Do-while loop

- Do-while is an exit controlled loop
  - *(while is entry controlled)*
  - When loop content should execute at least once
- E.g
  - Looping until user enters valid value
  - Gather game input for game loop

```
do
{
    //loop body
} while (condition);
```
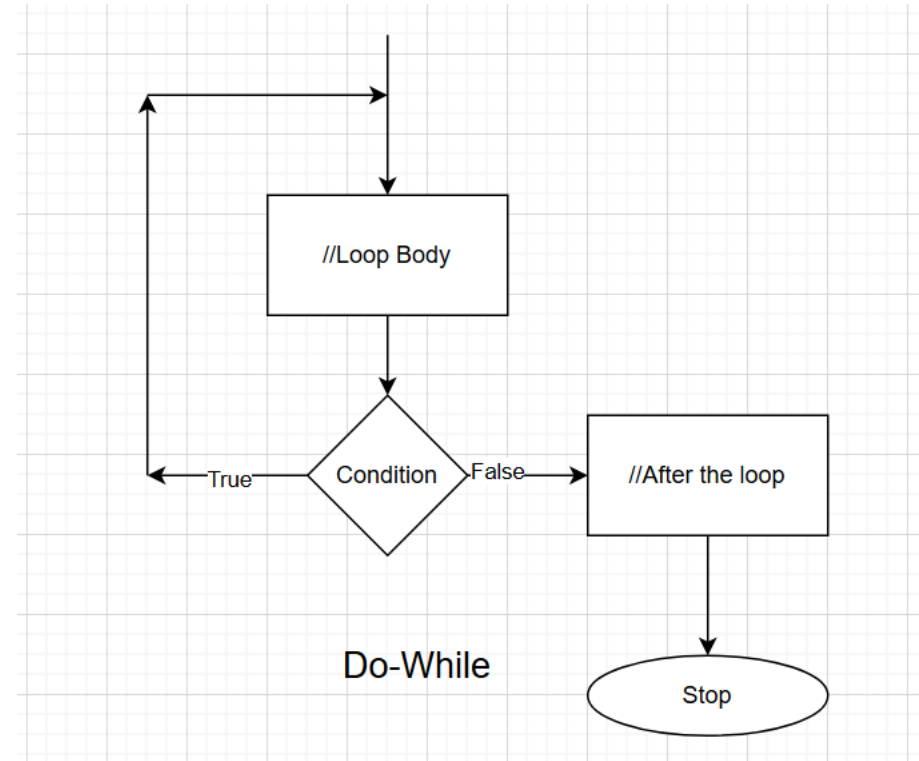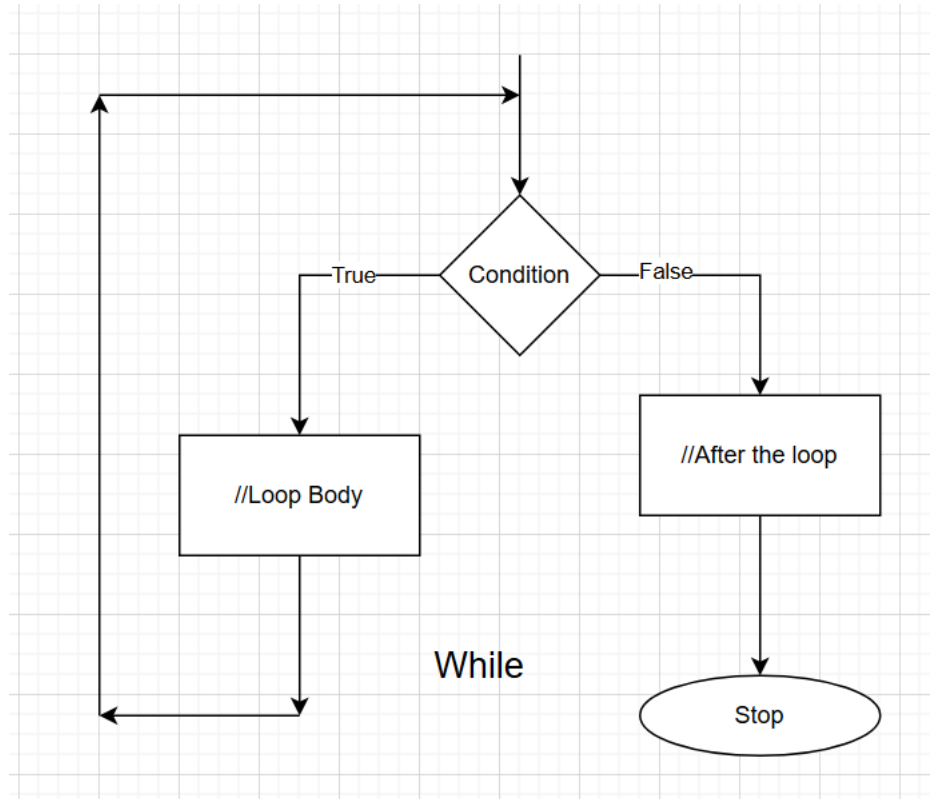
Do while : example

- Get a number between 1 and 4 from user.
  - Ask again if the numbers is not in range
  - Print the number and stop if it is in range.

```cpp
int num = 0;

do
{
    cout << "\n Enter a number between 1 and 4 :";
    cin >> num;
} while ( num<1 || num >4);

cout << "\n You entered : " << num;
```

While vs do-while, visualized

Jumps in loops

- Break
  - Jump to after the end of loop
  - Loop does not execute

```cpp
int num;
while (true) { // An infinite loop
    cout << "Enter a number (0 to stop): ";
    cin >> num;
    if (num == 0) {
        break; // Immediately exit the loop when the condition is met
    }
    //Process loop here
}
```

- Continue
  - Jump to the beginning of the loop
  - Loop continues to execute

```cpp
do
{
    cout << "\n this is even number detector";
    cin >> num;

    if (num % 2 != 0)
    {
        cout << "\n this is not an even number";
        continue; //Continue executing the loop
    }
    cout << "\n this is an even number";
} while (true);
```

Assignment 1:

Create a **menu** driven **calculator**

- Enter 2 numbers
- Get user option (0 to 5) repeatedly
  - 1 → add the two numbers and print
  - 2 → subtract 2nd from first number and print
  - 3 → multiply the two numbers and print
  - 4 →divide 2nd by first number and print
  - **5 →do nothing , and restart**
  - 0 →stop the loop
- Try using both loops and compare result
  - while
  - Do/while
- Use **break** and **continue**

Boss Assignment:
Menu driven multi purpose program

- Create a multi purpose program running until user enters 0
- Enter 2 numbers e.g. a, b;
- Get user options repeatedly and perform the operation
  - 1 → find the least common multiple of 2 numbers
  - 2 → find the highest common factor of the 2 numbers
  - 3 → find 'a' raised to the power of 'b' using loops : a^b
  - 4 → find $1/a^b$ using loops
  - 5 → do something random from between 1 and 4
    - #include<cstdlib >
    - int rand = 1+ rand()%4;
  - 0 → Exit the program

string data type

- We already know few built in types
  - int , float , char, bool
- string
  - Is a derived data type,
  - Useful for names, description etc.
- Header
  - #include<string>
- Usage
  - string name;
  - cin >> name;
  - cout <<name;