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

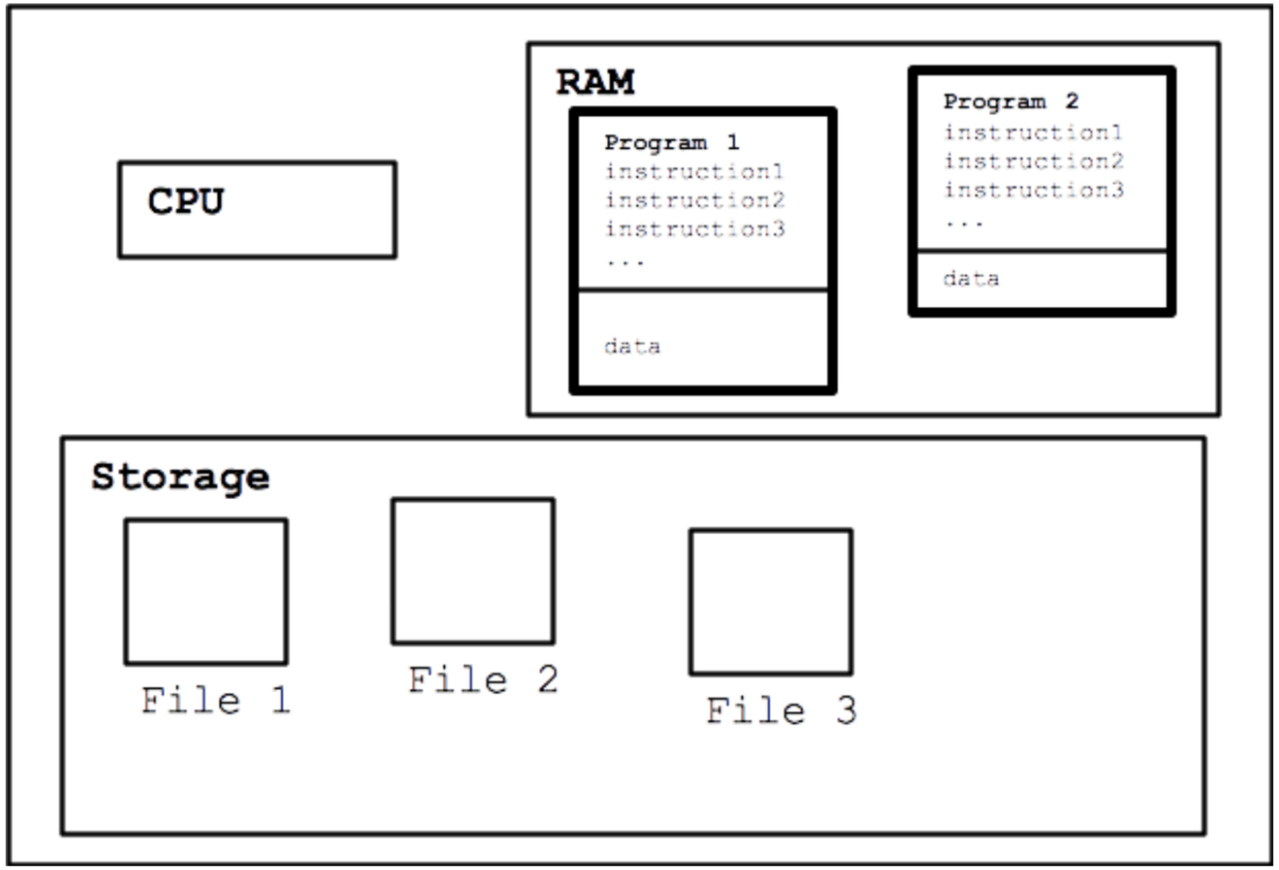
- Let’s go ,
- Program to print “Hello World”

```
-----
#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

For reference .... no need to memorize

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

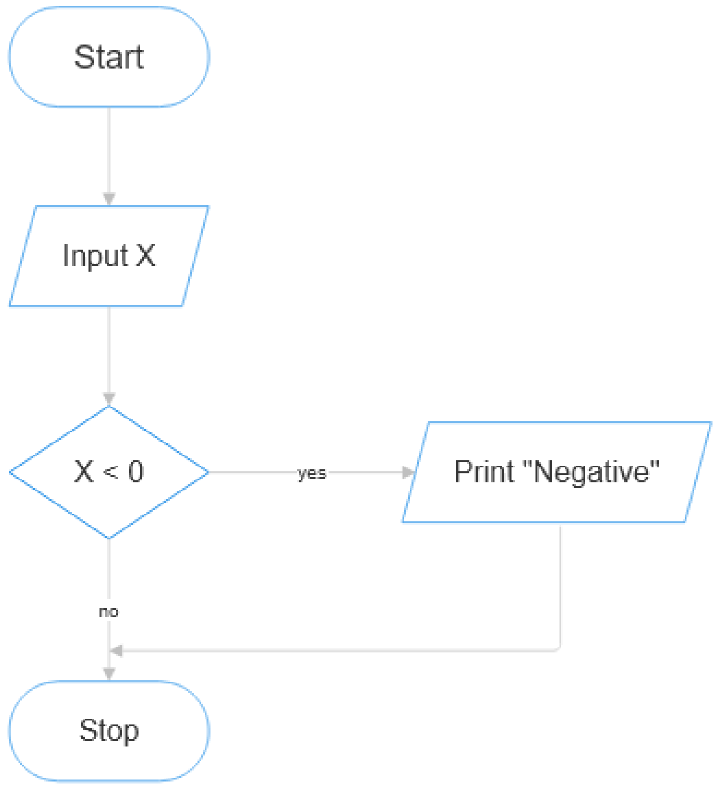
- Getting smaller
  - **Die** sizes have been shrinking
- Getting faster
  - Same size **die** have more **transistors**
- Getting crowded
  - More **core** counts per **die**
- Getting chatty
  - Networked , internet connected
- Getting efficient
  - More performance per unit of **power**

Back to C++, Decision Making & Branching

- if , else
- switch
- goto

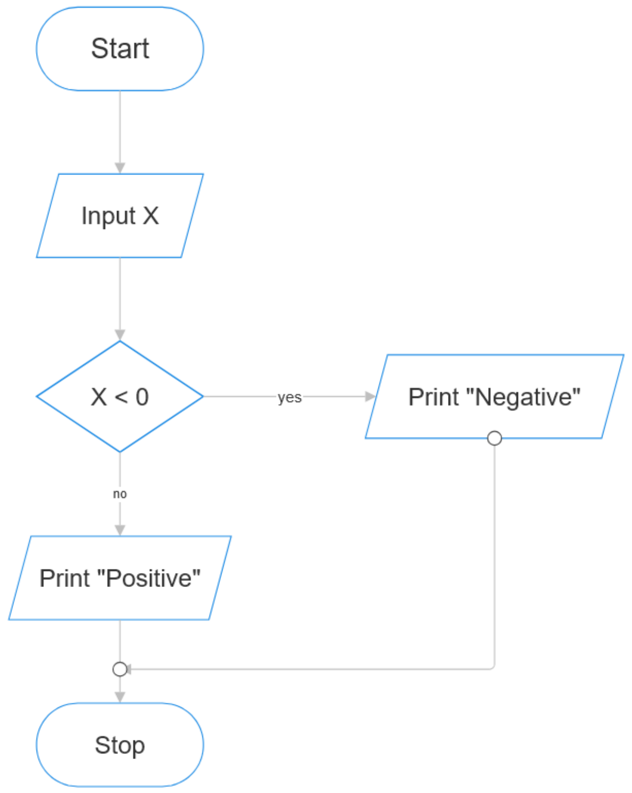
Making Decisions , using “if”

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

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

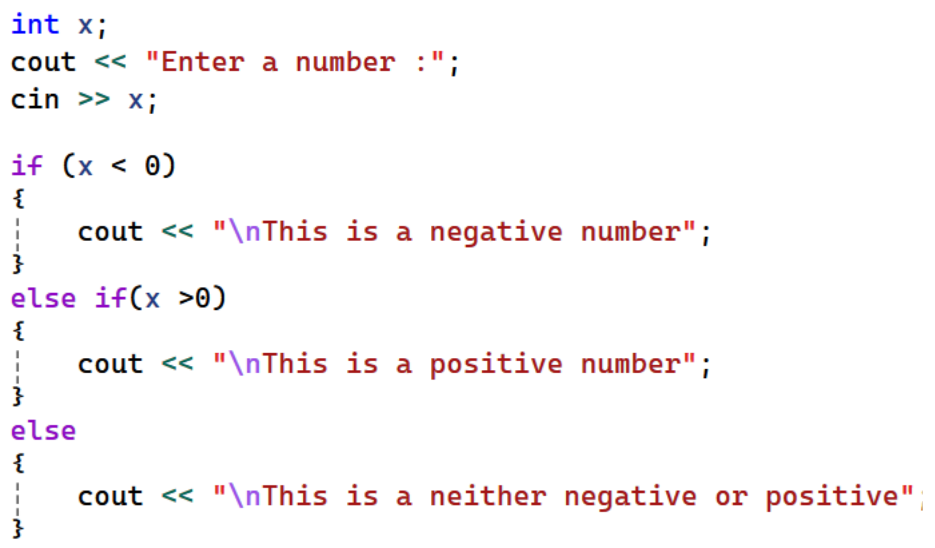


Making Decisions using “if else”

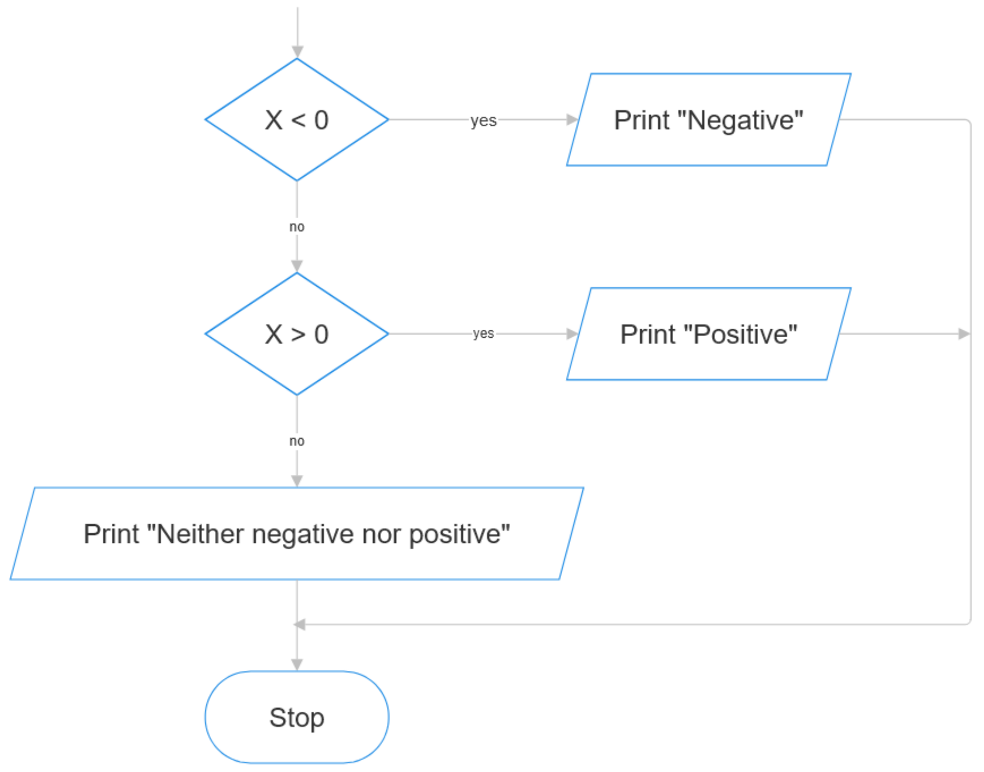
```
int x;
cout << "Enter a number :";
cin >> x;

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



Making decision, If else if else



```
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"
}
```

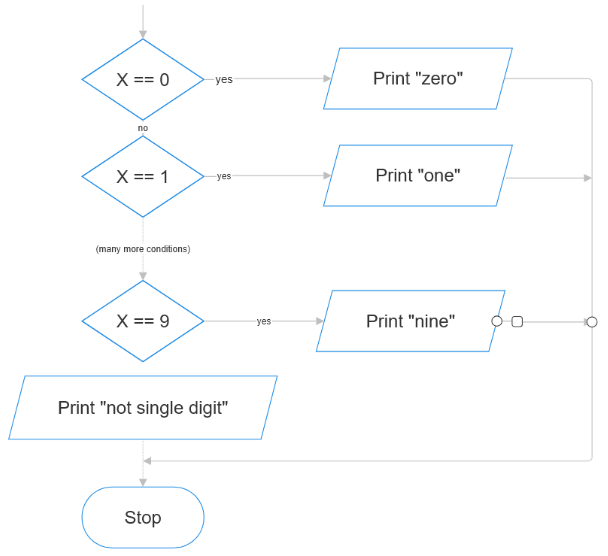


Relational operators

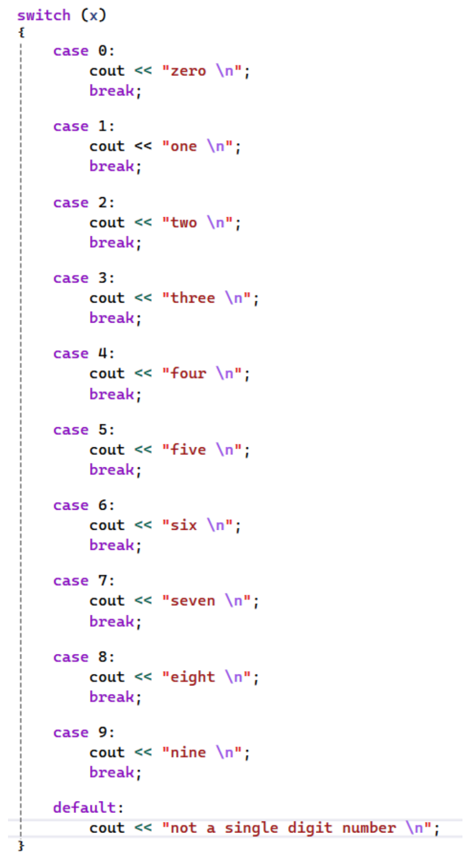
- less than
  - <
- greater than
  - >
- less then or equal to
  - <=
- greater than or equal to
  - >=
- is equal to
  - ==
- is not equal to
  - !=

Lab (if else)

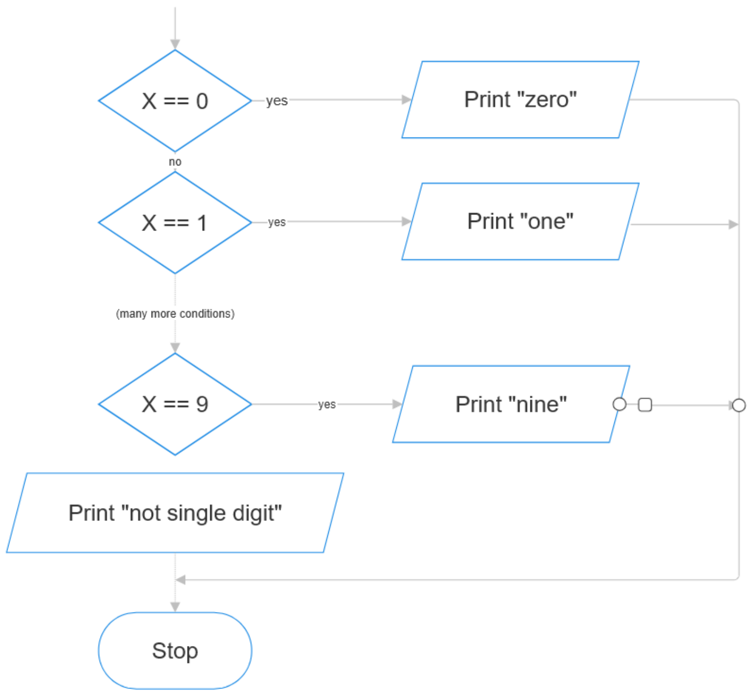
- Get user to enter a single digit number
- Print the digit in English
  - 0 → "zero"
  - 1 → "one"
  - …
  - 8 → "eight"
  - 9 → "nine"
  - (Anything else) "→ this is not a single digit number"



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 ?

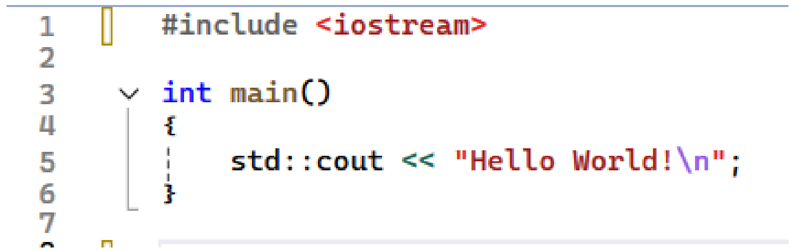
```
1. Seattle
2. Los Angeles
3. Washington DC
4. Chicago
Enter your response : _
```

Boss Assignment
Convert a string of 0,1 to decimal

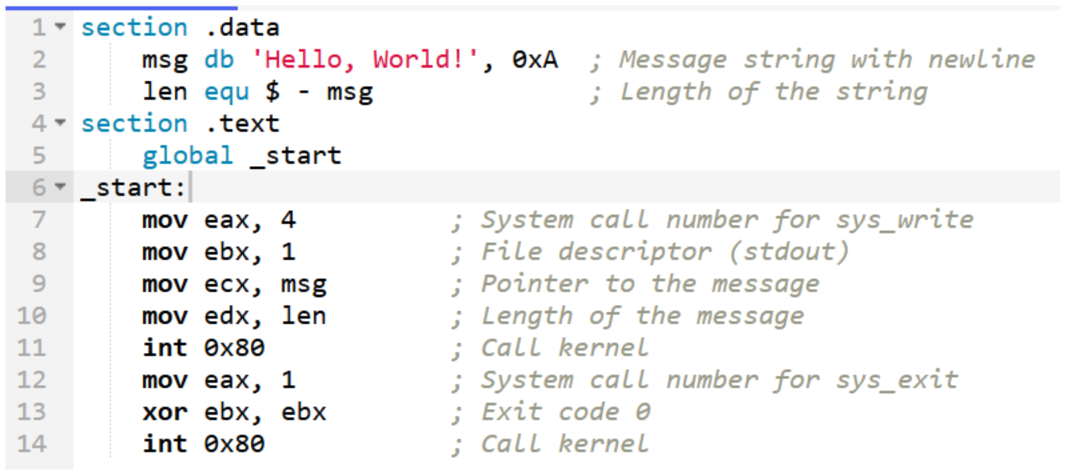
- Get a string of 0s and 1s from user
- Calculate what the decimal number for it is ?
- Note
  - This is very hard , may take hours/days/weeks
- Hint
  - You would need to know
    - string (or char array)
    - Loops (while)
    - Binary logic
    - *Loops within loops

Code, as seen by computer : HelloWorld

- Cpp vs assembly



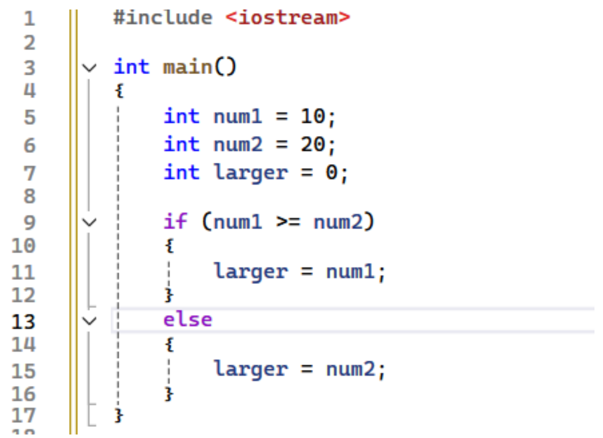
```cpp
#include <iostream>

int main()
{
    std::cout << "Hello World!\n";
}
```



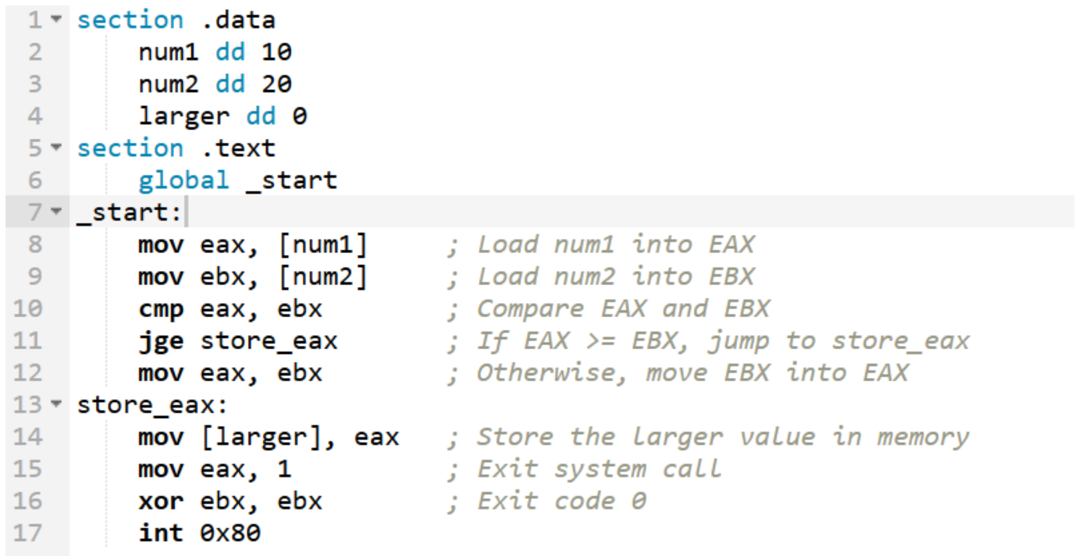
```nasm
section .data
    msg db 'Hello, World!', 0xA  ; Message string with newline
    len equ $ - msg              ; Length of the string
section .text
    global _start
_start:
    mov eax, 4               ; System call number for sys_write
    mov ebx, 1               ; File descriptor (stdout)
    mov ecx, msg             ; Pointer to the message
    mov edx, len             ; Length of the message
    int 0x80                 ; Call kernel
    mov eax, 1               ; System call number for sys_exit
    xor ebx, ebx             ; Exit code 0
    int 0x80                 ; Call kernel
```

Code , as seen by computer : Find larger num.

- Cpp vs Assembly



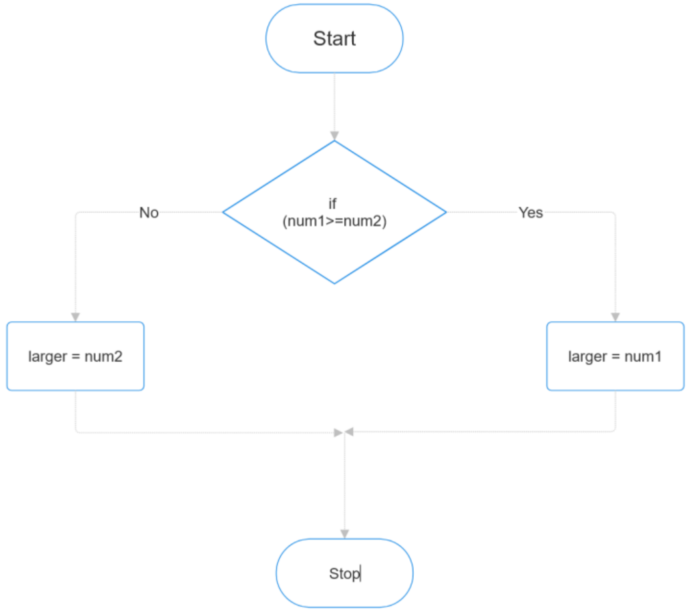
```cpp
#include <iostream>

int main()
{
    int num1 = 10;
    int num2 = 20;
    int larger = 0;

    if (num1 >= num2)
    {
        larger = num1;
    }
    else
    {
        larger = num2;
    }
}
```

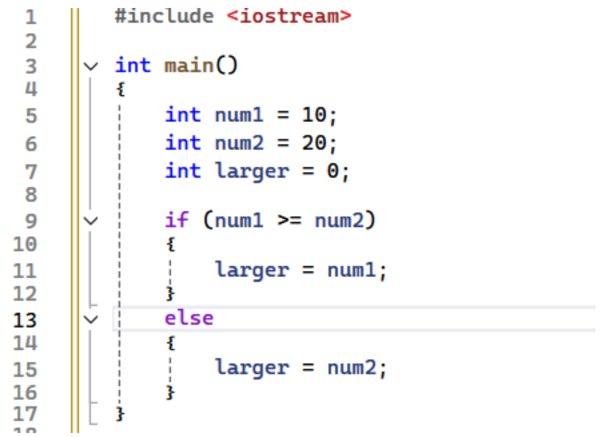


```asm
section .data
    num1 dd 10
    num2 dd 20
    larger dd 0
section .text
    global _start
_start:
    mov eax, [num1]      ; Load num1 into EAX
    mov ebx, [num2]      ; Load num2 into EBX
    cmp eax, ebx         ; Compare EAX and EBX
    jge store_eax        ; If EAX >= EBX, jump to store_eax
    mov eax, ebx         ; Otherwise, move EBX into EAX
store_eax:
    mov [larger], eax    ; Store the larger value in memory
    mov eax, 1           ; Exit system call
    xor ebx, ebx         ; Exit code 0
    int 0x80
```

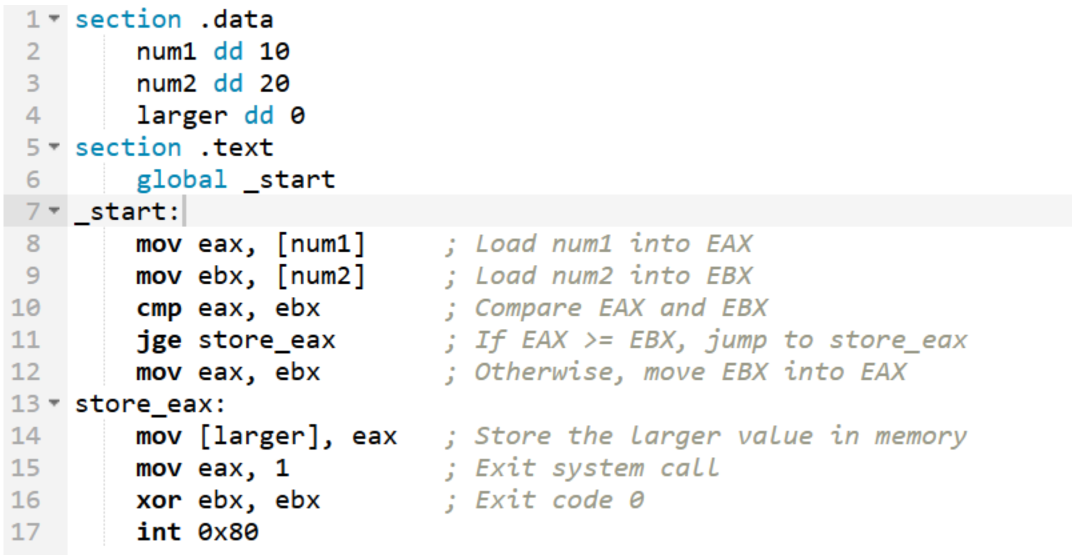
Branching visualized

- Code looks sequential
- But really has branches





```
#include <iostream>

int main()
{
    int num1 = 10;
    int num2 = 20;
    int larger = 0;

    if (num1 >= num2)
    {
        larger = num1;
    }
    else
    {
        larger = num2;
    }
}
```



```
section .data
    num1 dd 10
    num2 dd 20
    larger dd 0
section .text
    global _start
_start:
    mov eax, [num1]       ; Load num1 into EAX
    mov ebx, [num2]       ; Load num2 into EBX
    cmp eax, ebx          ; Compare EAX and EBX
    jge store_eax         ; If EAX >= EBX, jump to store_eax
    mov eax, ebx          ; Otherwise, move EBX into EAX
store_eax:
    mov [larger], eax     ; Store the larger value in memory
    mov eax, 1            ; Exit system call
    xor ebx, ebx          ; Exit code 0
    int 0x80
```

Loops : Things that happen over and over …

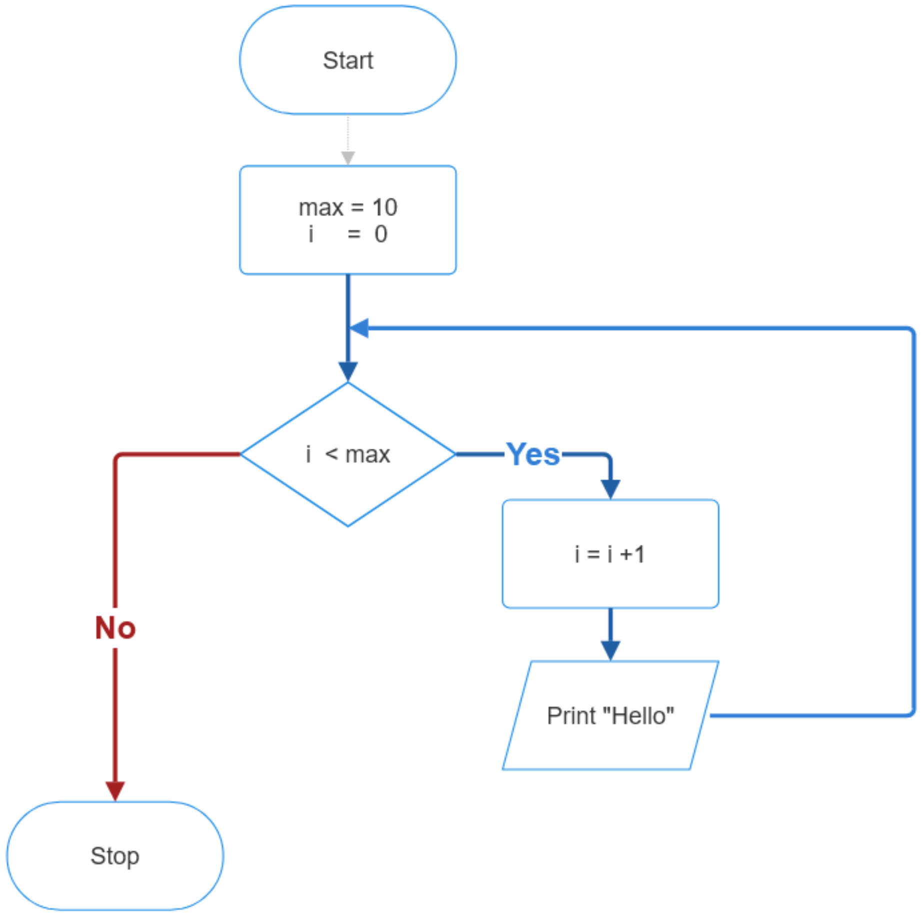
- Years
  - …,2024,2025, …
- Seasons
  - Spring, summer, fall, winter
- School
  - Learn , Homework, Tests
  - Scoring
- Reading a book
  - Title,… page, page,page …,the end

Decision making and looping

- Loops
  - Print “Hello”10 times
  - 10 x cout ?
- What if 100 times ?
  - 100 x cout ??
- Can we do better ?

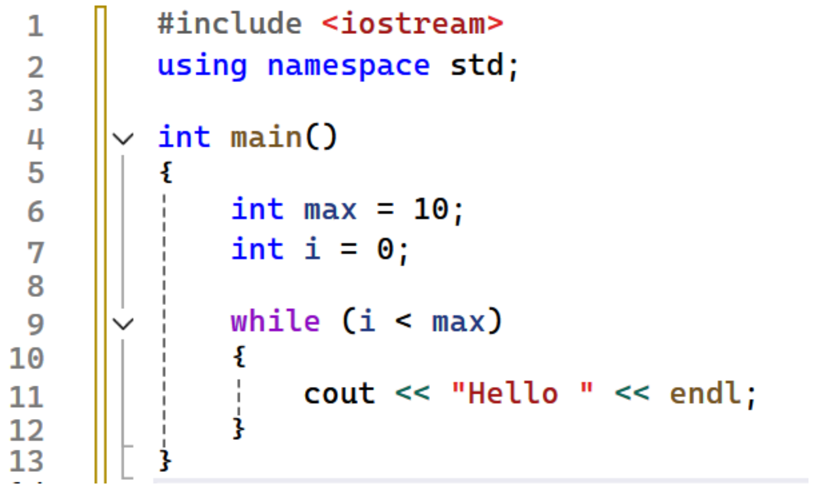
Loop : visualized

- Print “Hello”10 times

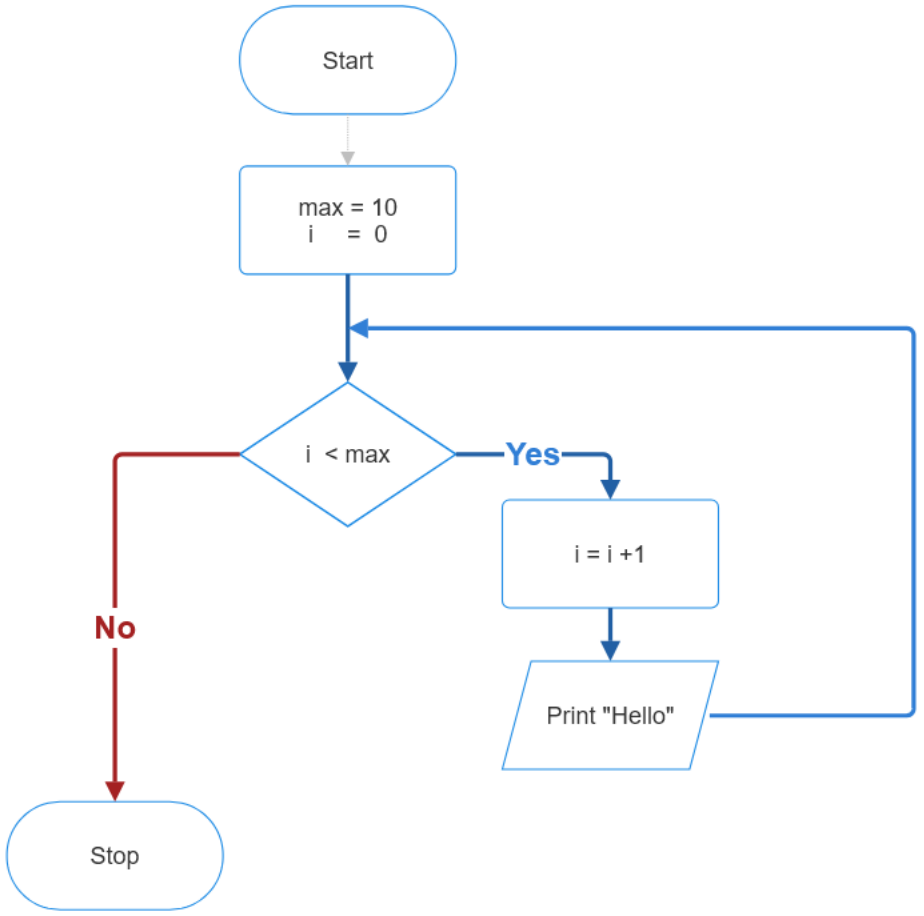


Loop : visualized

- Print “Hello”10 times

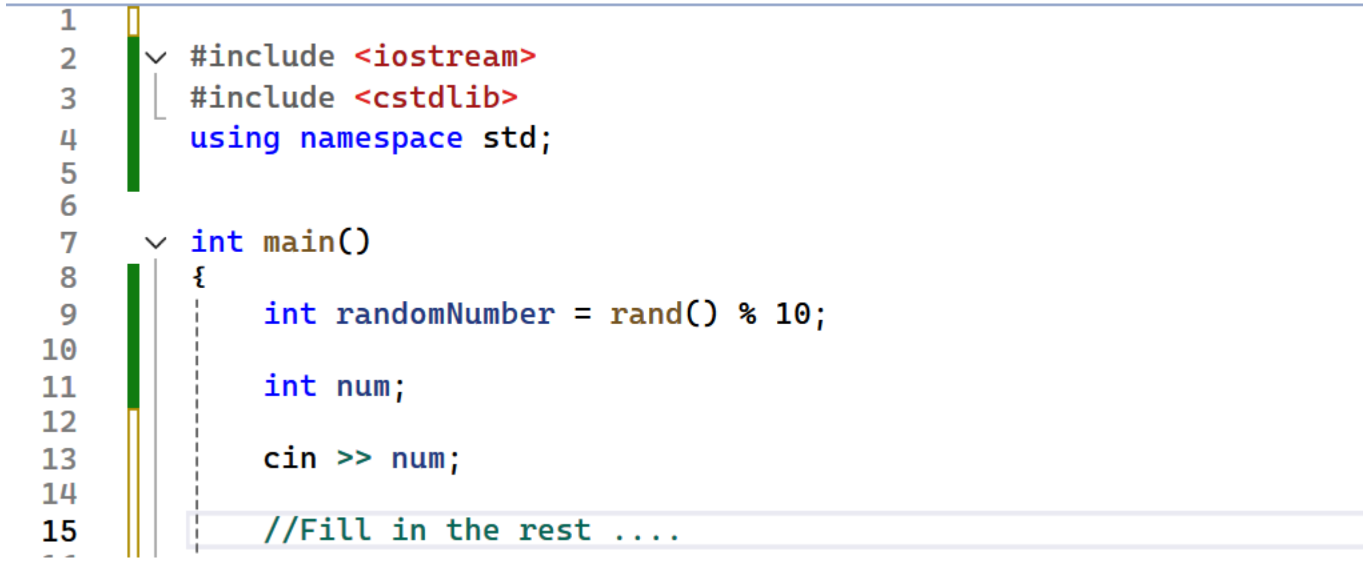


```
#include <iostream>
using namespace std;

int main()
{
    int max = 10;
    int i = 0;

    while (i < max)
    {
        cout << "Hello " << endl;
    }
}
```



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)



```

#include <iostream>
#include <cstdlib>
using namespace std;


int main()
{
    int randomNumber = rand() % 10;

    int num;

    cin >> num;

    //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 $\rightarrow$ 2
    - 6,9 $\rightarrow$ 3
- EASYMEDIUMHARD

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;