

IC100: Introduction to Computing

Course Logistics

Instructor Details

Dr. Sk Subidh Ali

Assistant Professor

Department of Computer Science and Engineering

Office: Room No. 206, Academic Building

Email: subidh@iitbhilai.ac.in

The Course

- The course introduces you to programming using the C programming language
- No prior exposure to programming is needed
- You will be evaluated on both theoretical and practical aspects of programming, using written exams and labs

Why am I Doing This Course?

- Every discipline uses computing: All branches of engineering, sciences, design and arts.
- Understand how computers work
- Write your own programs
 - Automate boring repetitive stuff!

Lectures

- Lectures

- Mon, 08:30 am – 09:50 am , 107
- Wed, 11:30 am 12:50 pm, 107

- Location

- Room 107

Labs

- Lab is divided into 5 Batches:
 - Batch I, Batch II, Batch III, Batch IV, Batch V
- Schedule: 2:00 PM – 6:20 PM
 - Batch I : Monday
 - Batch II : Wednesday
 - Batch III: Friday
 - Batch IV : Thursday
 - Batch V : Saturday
- Location:
 - Room no 109
 - (CC Lab) (Please make sure you carry id card to enter to the lab)
- Labs start From 31st oct (Today)
 - First week labs are meant to familiarize you with the lab workflow

Tutorials

- There will be tutorials every Friday
 - 11:30 AM - 12:50 PM
- There will be 4 batches for tutorials
 - Batch I, Batch II, Batch III, Batch IV
- You can ask questions and clarify doubts regarding lecture material
- Examples illustrating lecture material will be covered
- Quizzes will be held during the last 15 minutes of every tutorial sessions

Labs

- Sunday: Could be used to make up for lab days lost due to holidays.
- There will be Teaching Assistants (TAs) to help in the labs.
- In each lab, you will be given a few problems to solve.
 - Students must work on their own.
 - Discussion is allowed, but **sharing of code in any form is NOT permitted.**

Lab Assignments

- Lab assignment will be posted on the day of the lab, at 2 PM
- It has to be submitted by 6:20 PM
- There will additional question carrying bonus marks

Weightage (Theory)

- Quizzes: 10%
 - Conducted in lecture and tutorial hours
- Tierce1: 10%
- Tierce2 : 15%
- Tierce3: 15%
- Homework: 3%
- Classroom Assesment: 5%

Weightage (Lab)

- Labs: 12%
 - Weightage of later labs may be more. (First lab: 0 weight)
 - **No make-up** lab for absentees.
- Lab exams: 30%
 - Tierce1: 10%
 - Tierce2: 10%
 - Tierce3: 10%

Semester Workload

- Every week
 - 3 hours lectures
 - 1.5 hour tutorial
 - 4.5 hours lab work
- Semester
 - 30 Quizzes
 - 3 lab exams
 - 3 theory exams

Copying

- Copy at your own risk
 - in any component (lab/quiz/exams/lab exams)
 - If you are caught, you get penalised on grade (most likely F)
 - Will not be allowed to drop the course
 - Case reported to DoAA
 - No warning or second chance
 - All parties involved in copying will be held equally responsible. Copying from internet is penalized equally
 - Policy may change on need basis

Copying

- Read-protect your directories so that others cannot copy from your directory
- Do not share your codes with friends
- Do not leave printouts, notes etc. containing your code unattended

Course Page and Website

- Every Website
 - <https://iit-bhilai-ic100.github.io/> (Will be up in some time)
- Course Page
 - A mail will be sent to you for enrolling in piazza where you can submit assignments, homeworks and have discussions.

Course Materials

- Every student must register this course in Piazza.
- All course materials, including lectures, exam solutions, quiz solutions etc., will be posted on Piazza and subsequently on course web site.
- Use piazza for interaction
 - Allows instructor, tutors and your classmates to answer any issues

Textbooks

- There are several books available on C programming language
 - C in Depth by Deepali Srivastava
 - The C Programming Language by Kernighan and Ritchie, Prentice-Hall India
 - Programming in ANSI C by Balaguruswamy
 - Any other standard book on C would also be good
- It is recommended that you have a book and refer to it throughout the semester and beyond

E-Mailing

- Please make sure you mention your section and roll number in emails
 - While Submission don't forget to mention name, roll number and batch of the lab.
 - Prefer using discussion feature of Piazza

Today's Class

- What is programming?
- What is the programming workflow?
- What do simple C programs look like?

What is a Computer ?



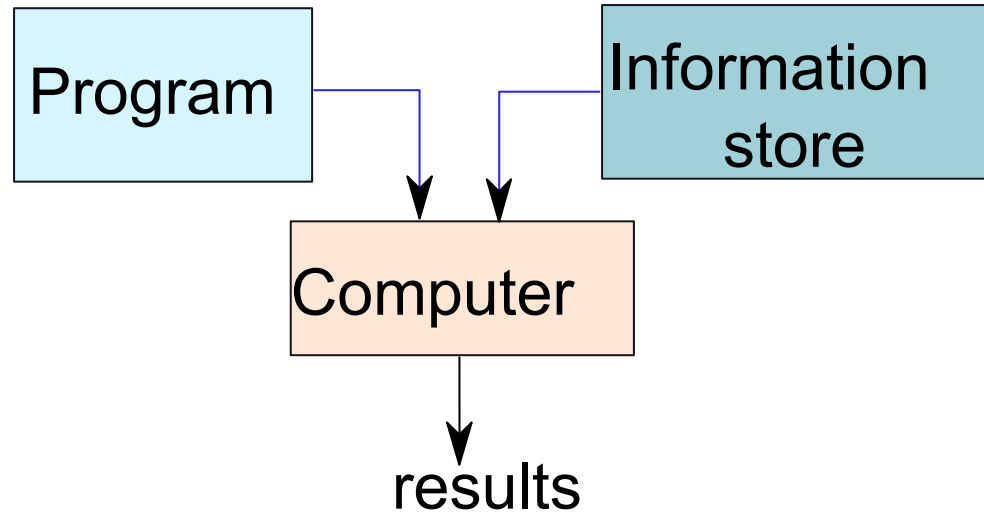
A computer is a general purpose device that can be programmed to process information, and yield meaningful results.



What is a Computer Program?

- **Computer:** Device which can be instructed to carry out arithmetic and logical operations automatically
 - Must take inputs
 - Must store inputs
 - Must process input
 - Must give outputs
- How would the Computer process the inputs?
 - Needs precise instructions
 - PROGRAM
- We will learn programming in this course

How does it work ?



- Program – List of instructions given to the computer
- Information store – data, images, files, videos
- Computer – Process the information store according to the instructions in the program

Memory



Memory

Hard disk

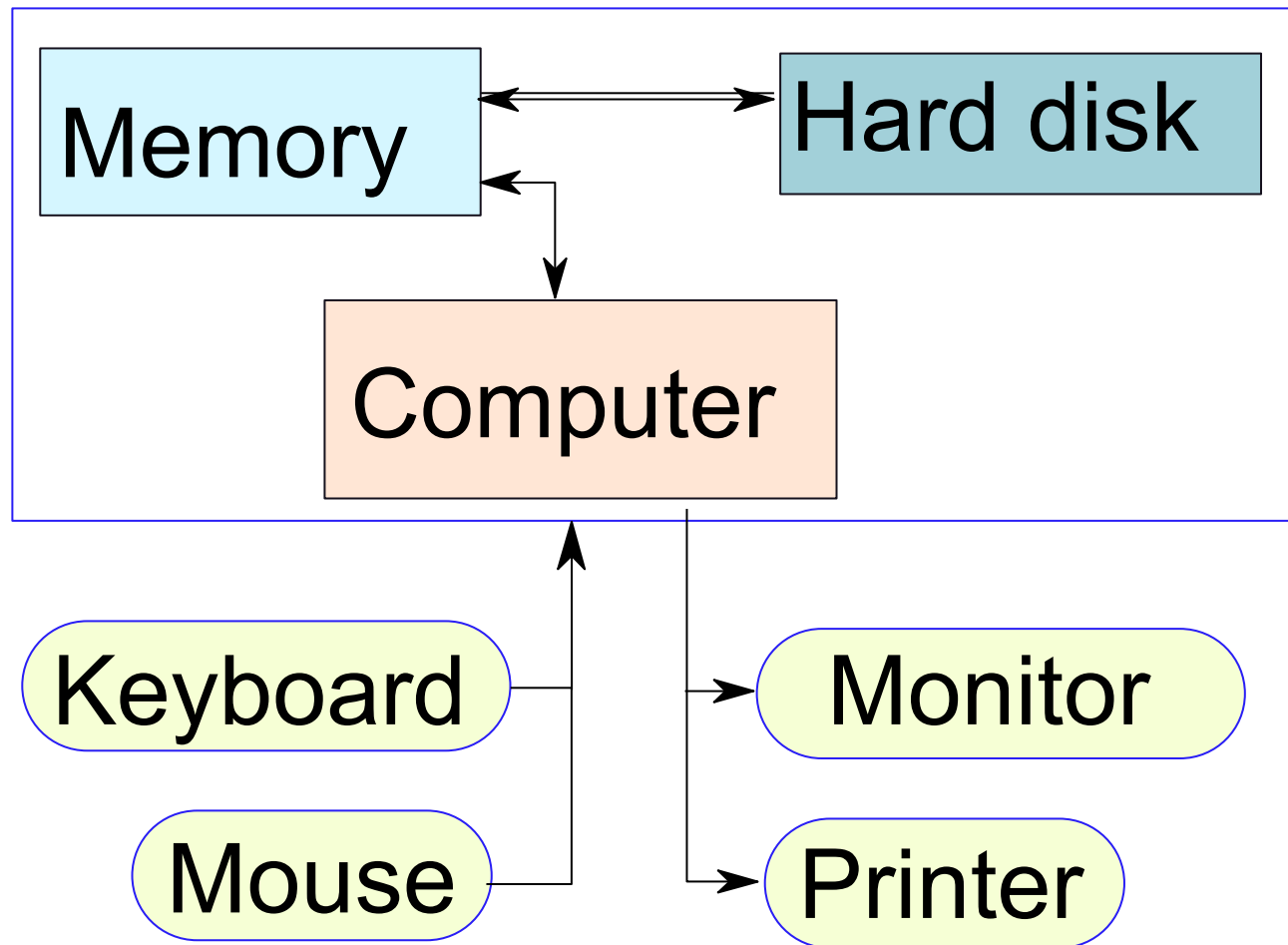


Computer



- Memory (primary memory) – Stores programs and data. Temporary storage
- Hard disk – stores programs/data permanently

Let us make it a full system ...



Process of Programming: Modelling

- Define and model the problem.
 - In real-life this is important and complicated.
- Modelling the grade of students in a class
 - Explicit Input list of people with attributes
 - Roll number
 - Name
 - Marks obtained
 - Implicit inputs modelling
 - List of grades (A+,A,B,C, F)
 - Range of marks for each grad (Ex. 90-99 =>A+, 80-89 =>A)
 - Sort student marks
 - Apply grade
 - Output
 - List of students with their grade

Process of Programming: Algorithm

- Obtain a logical solution to your problem.
- A logical solution is a finite and clear step-by-step procedure to solve your problem.
- Also called an Algorithm (or *recipe*).

Algorithms for grading

- Input:
 - List of students with name, roll number and marks obtained
 - Mapping between grade and marks range
- Instructions
 - Input student details
 - Extract marks from the input details
 - Search the range in which input mark lies
 - Output the corresponding grade

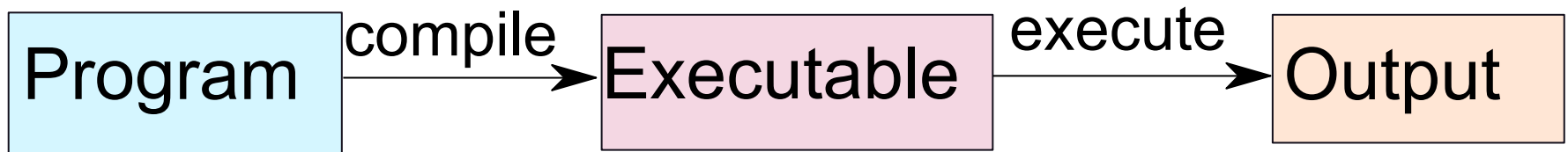
Write the algorithm for computing GCD of two numbers



GCD

1. An algorithm to find the greatest common divisor of two positive integers m and n , $m \geq n$.
2. The ugly way
 - a. Take the smaller number p .
 - b. For each number k , $p \geq k \geq 1$, in descending order, do the following.
 - i. If k divides m and n , then k is the gcd of m and n

How to Instruct a Computer ?

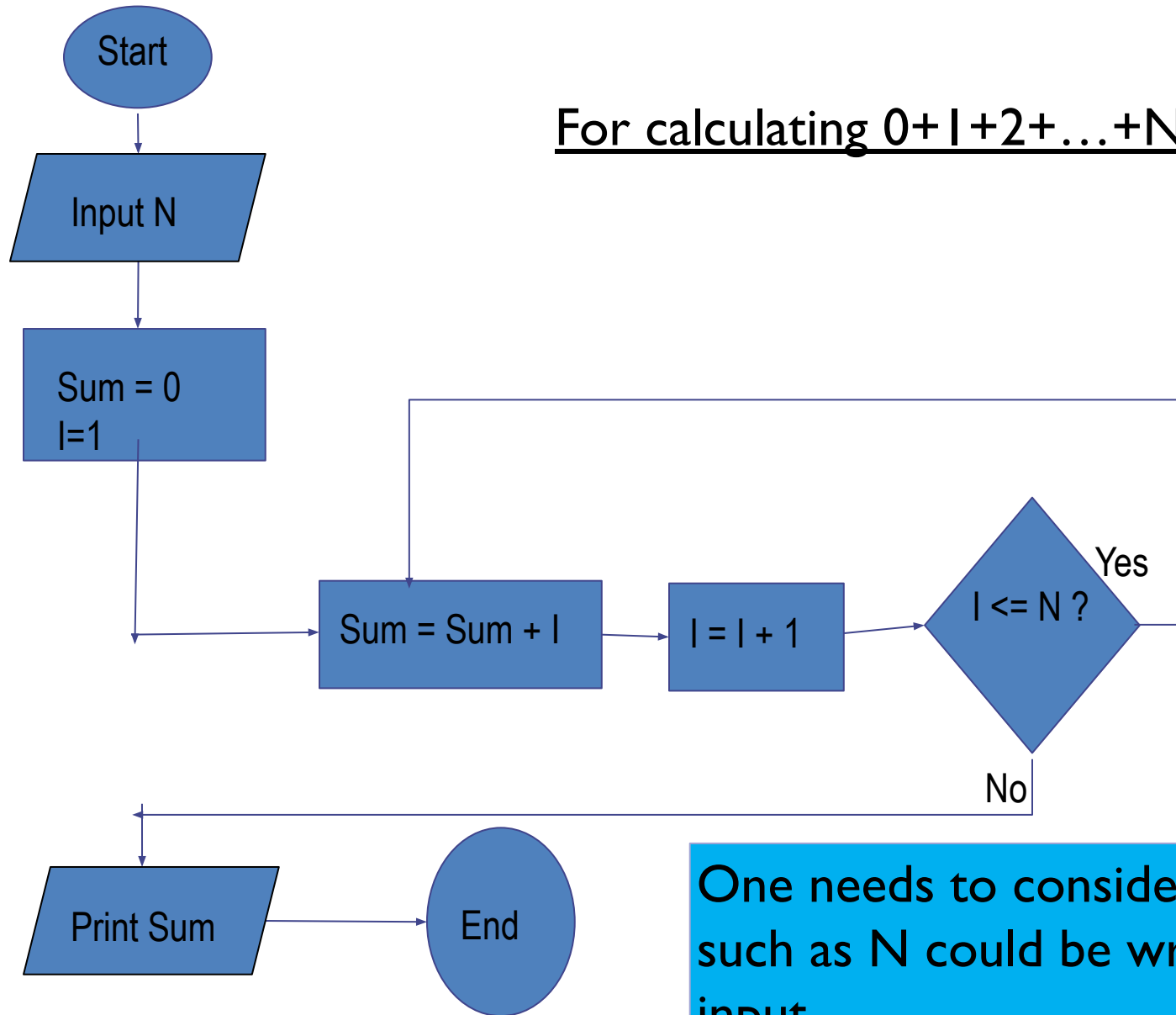


- * Write a program in a high level language – C, C++, Java
- * **Compile** it into a format that the computer understands
- * Execute the program

Flow Chart

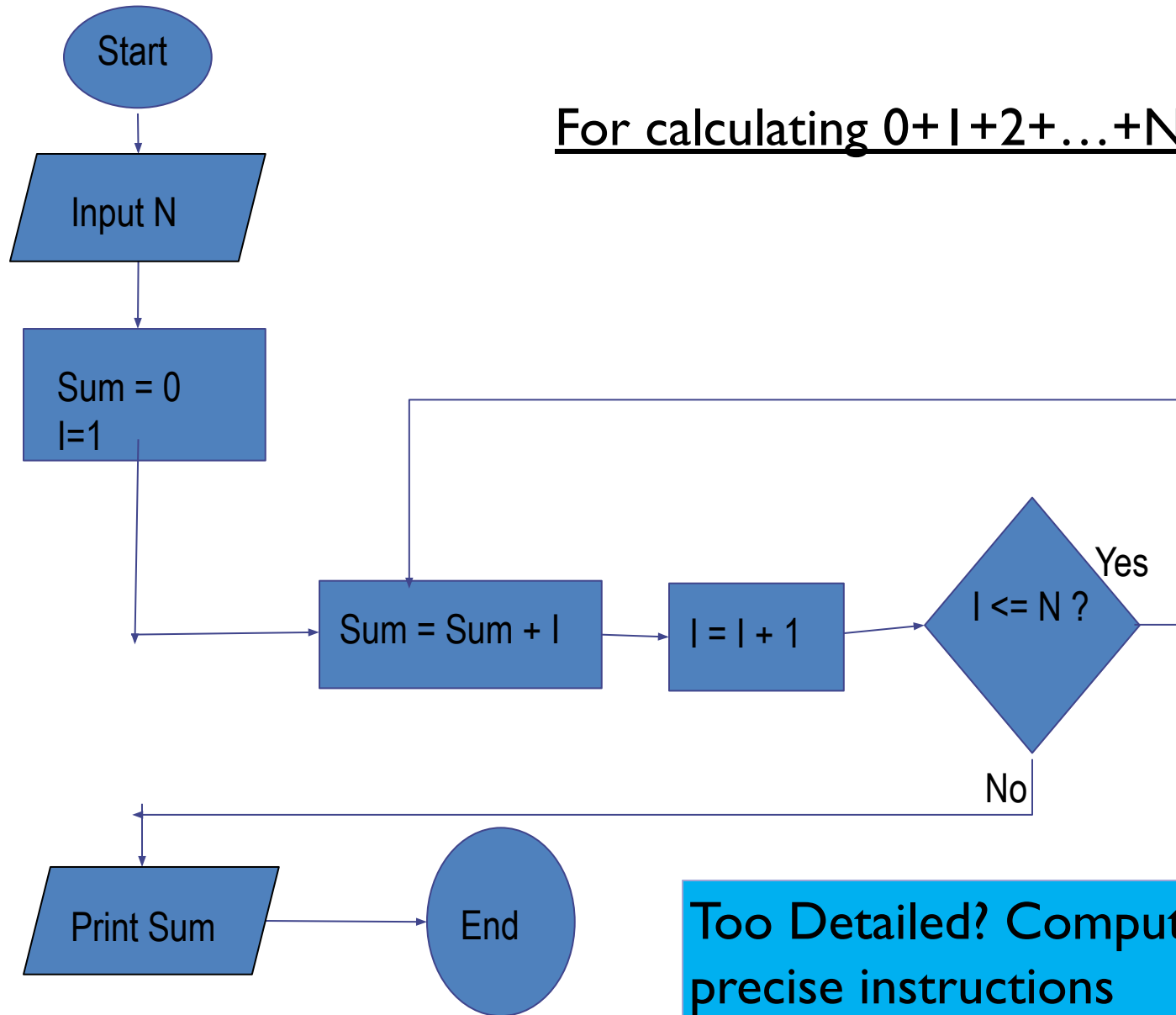
- We can visualize an algorithm using a Flowchart.
- Very important step in the programming process.

Flowchart to Depict Algorithm



One needs to consider all cases, such as N could be wrongly input

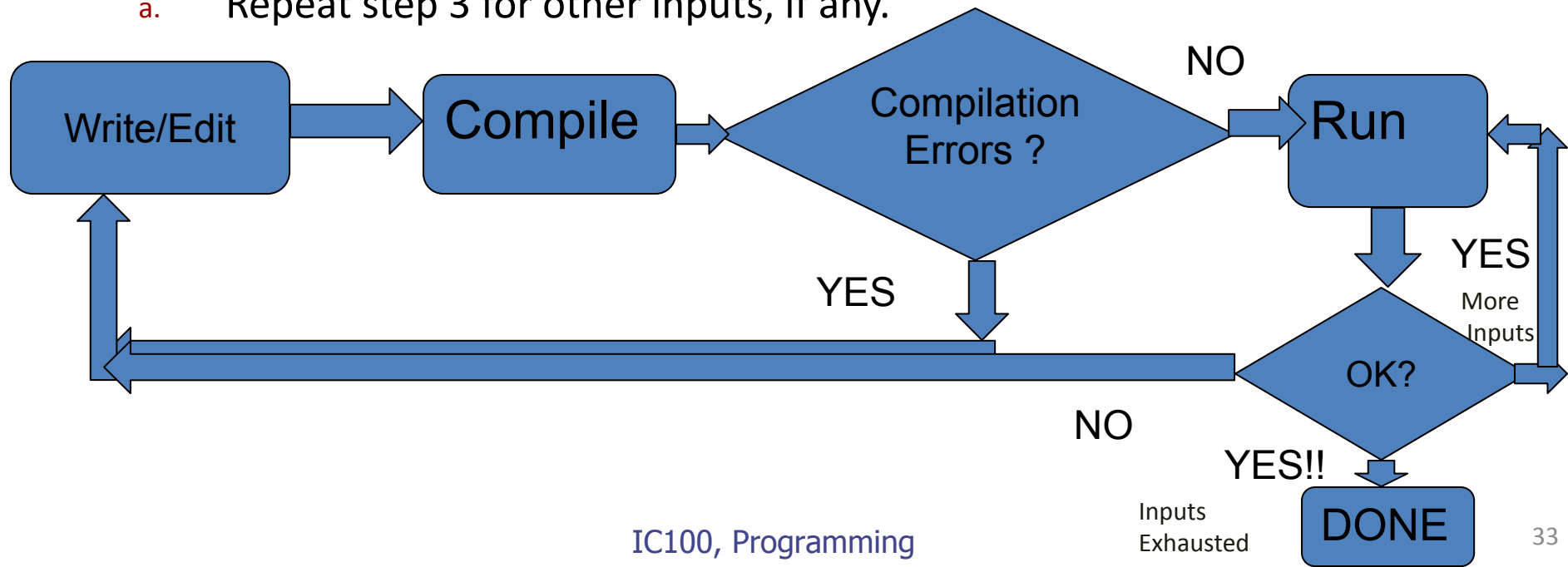
Flowchart to Depict Algorithm



Too Detailed? Computers need precise instructions

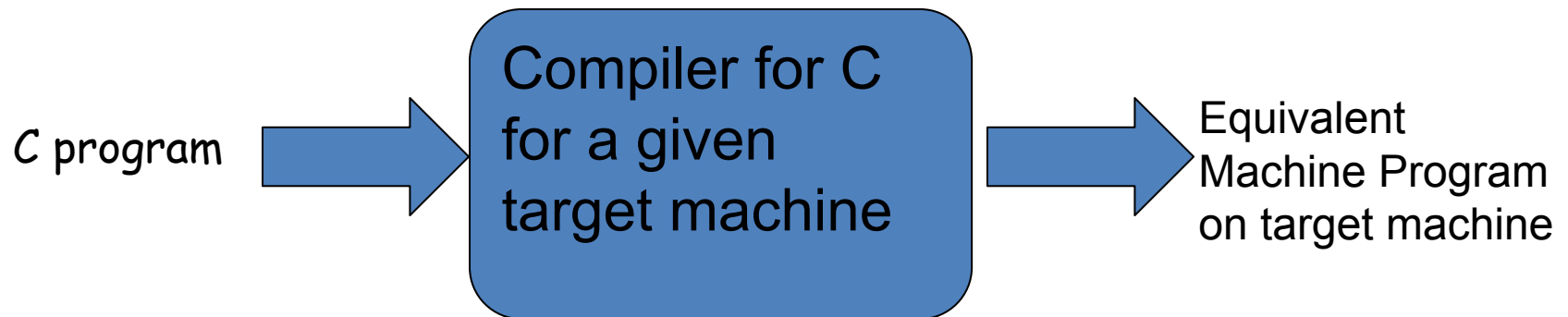
The Programming Cycle

1. Write your program or **edit** (i.e., change or modify) your program.
2. **Compile** your program. If compilation fails, return to editing step.
3. **Run** your program on an input. If output is not correct, return to editing step.
 - a. Repeat step 3 for other inputs, if any.



Why program in high level languages like C

- Writing programs in machine language is long, tedious and error-prone.
- They are also not portable—meaning program written for one machine may not work on another machine.
- Compilers work as a bridge.
- Take as input a C program and produce an equivalent machine program.



IDE for Edit-Compile-Run cycle

- In this course we will be using gedit or simple text editor in ubuntu to write the programs.
- First login to the system.
- The labs in the first week will introduce you to the system in more detail.

Simple Program

- In the first week, we practice the simplest C programs.

```
# include <stdio.h>
int main () {
    printf("Welcome to IC100");
    return 0;
}
```

The program prints the message "Welcome to ESC101"

Program Components

```
# include <stdio.h>
```

```
int main ()
```

```
{
```

```
    printf("Welcome to IC100");
```

```
    return 0;
```

```
}
```

1. This tells the C compiler to include the standard input output library.

2. Include this line routinely as the first line of your C file.

printf is the function called to output from a C program. To print a string, enclose it in " " and it gets printed.

"return" returns the control to the caller (program finishes in this case.)

main() is a function. All C programs start by executing from the first statement of the main function.

printf

- printf is the “voice” of the C program
 - Used to interact with the users
- printf prints its arguments in a certain format
 - Format provided by user

More printing ... (code and see)

```
#include <stdio.h>
void main()
{
    printf ("Hello, World! ");
    printf ("Hello \n World! \n");
}
```

Some more printing

```
#include <stdio.h>
void main()
{
    printf ("Hello, World! \n");
    printf ("Hello \n World! \n");
    printf ("Hell\no \t World! \n");
}
```


Another Simple Program

- Program to add two integers (17 and 23).

```
#include <stdio.h>
int main () {
    int a = 17;
    int b = 23;
    int c;
    c = a + b;
    printf("Result is %d", c);
    return 0;
}
```

The program prints the message: **Result is 40**

```
# include <stdio.h>
```

```
int main () {
```

```
    int a = 17;
```

```
    int b = 23;
```

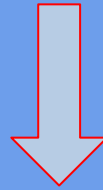
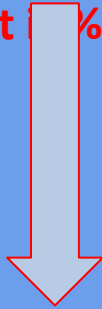
```
    int c;
```

```
    c = a + b;
```

```
    printf("Result %d",c);
```

```
    return 0;
```

```
}
```



This tells the compiler to reserve a "box" large enough to hold an integer value. The box is named "a" for use in the rest of the program.

+ is an operator used to add two numbers. The numbers come from the values stored in the boxes named "a" and "b"

%d tells printf to expect one integer argument whose value is to be printed. We call it placeholder. We will see more placeholders soon.

"= 17" stores value 17 in the box that we have named "a". It is OK to skip this part and store value later as we do for box named "c".

Do you now understand this program?

- Program to add two integers (17 and 23).

```
#include <stdio.h>
int main () {
    int a = 17;
    int b = 23;
    int c;
    c = a + b;
    printf("Result is %d", c);
    return 0;
}
```

The program prints the message: **Result is 40**

Reading values from keyboard

```
#include <stdio.h>
void main()
{
    int num ;
    scanf ("%d", &num) ;
    printf ("No. of students is %d\n", num) ;
}
```

Centigrade to Fahrenheit

```
#include <stdio.h>
void main()
{
    float cent, fahr;
    scanf("%f",&cent);
    fahr = cent*(9.0/5.0) + 32;
    printf( "%f C equals %f F\n", cent, fahr);
}
```

Next class

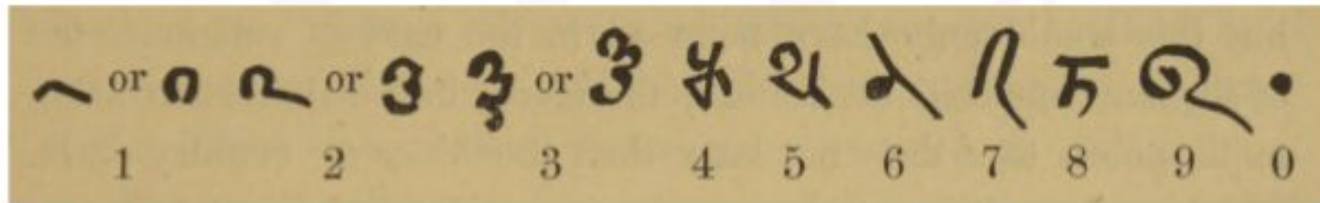
- The alphabet of C
 - Like int
- The grammar of C
 - Like using “ ” to delimit strings
- The keywords of C
 - Like printf

ACKNOWLEDGEMENTS

Slides were prepared from resources shared by Prof Rajat Moona and Prof. Amey Karkare for the course ESC101 in IIT Kanpur.

Fundamentals: Numbers

Indian System



Bakshali numerals, 7th century AD

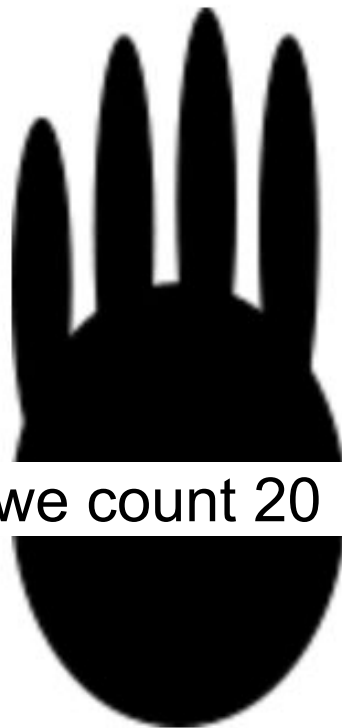
- Uses the place value system

$$5301 = 5 * 10^3 + 3 * 10^2 + 0 * 10^1 + 1 * 10^0$$

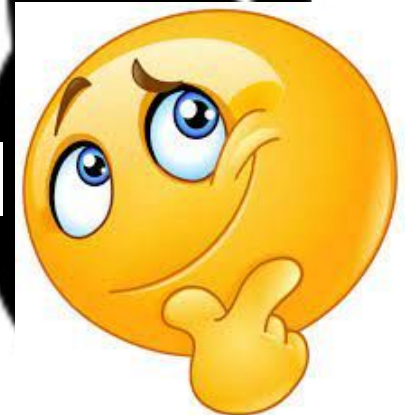
Example in base 10

Number Systems in Other Bases

- Why do we use base 10 ?
 - because ...

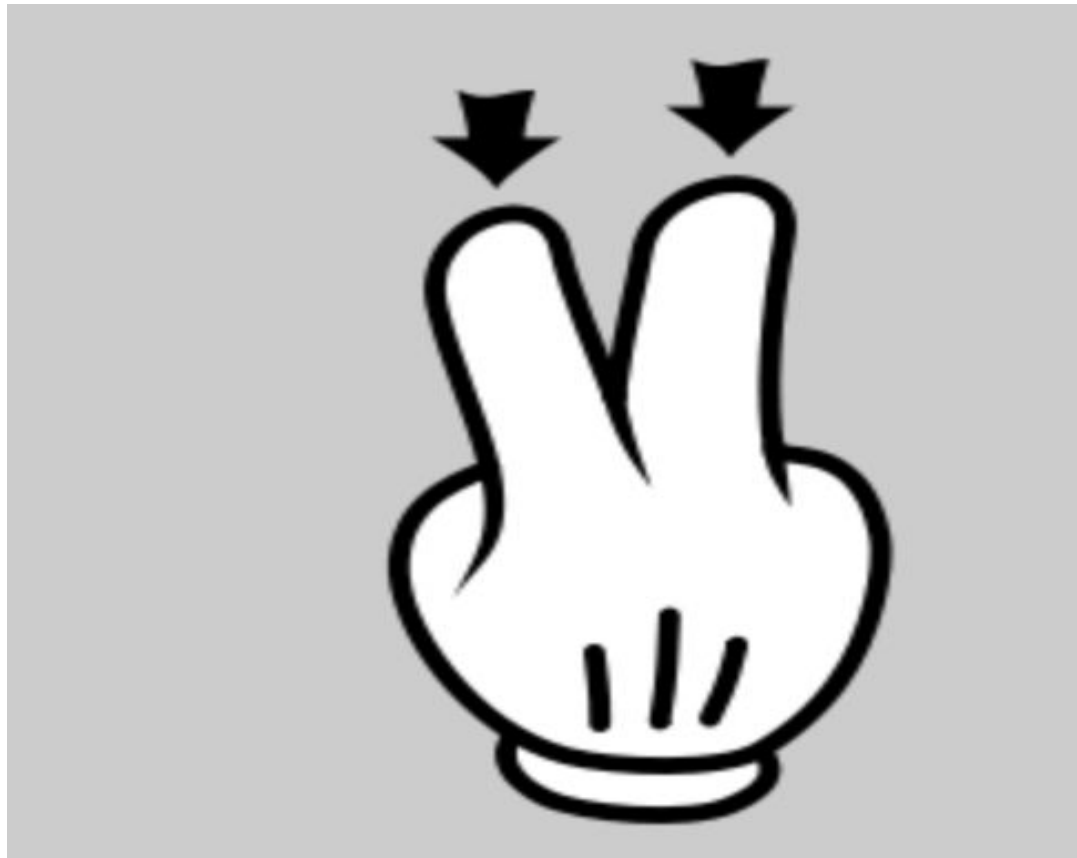


How do we count 20 using 10 fingers?



What if we had a world in which ...

- People had only two fingers.



Representing Numbers

$$1023_{10} = 1 * 10^3 + 0 * 10^2 + 2 * 10^1 + 3 * 10^0$$

Decimal

base

$$17_{10} = 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 =$$
$$10001_2$$

Binary

Base = 2

Binary Number System

- They would use a number system with base 2.

Number in decimal	Number in binary
5	101
100	1100100
500	111110100
1024	10000000000

Hexadecimal and Octal Numbers

- Hexadecimal numbers
 - Base 16 numbers – 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
 - Start with 0x
- Octal Numbers
 - Base 8 numbers – 0,1,2,3,4,5,6,7
 - Start with 0

Examples

Decimal	Binary	Octal	Hexadecimal
9	1001	0 11	0x 9
12	1100	0 14	0x C
17	10001	0 21	0x 11
28	11100	0 34	0x 1C



Convert 110010111 to the octal format : = 0612

Convert 111000101111 to the hex format : = 0xC2F

Bits and Bytes

- * Computers do not understand natural human languages, nor programming languages
- * They only understand the language of **bits**

Bit	0 or 1
Byte	8 bits
Word	4 bytes
kiloByte	$1024 = 2^{10}$ bytes
megaByte	2^{20} bytes