# Introduction to Computer Programming

Recompiled -
Dr. Jaideepsinh Raulji
Assistant Professor,
CSE, SET,
Navrachana University

# Computer Programming

Where do you find Programs ?

- Smart Phones (Applications)
- Computer Systems
- Electronic Home Appliances like Washing Machines, Ovens etc.
- Electronic Gadgets  (Firmware)

# Computer Programming

Could you list types of Programs ?

- Operating System
- Drivers
- Application Softwares
- Web-Sites
- Web-Services
- DBMS
- And so many

# Computer Programming

So how to build such Programs ?

- Using Programming languages like
- C, C++, Java, C#, VB, Python etc.

# Computer Programming

What can we do with Programming Languages ?

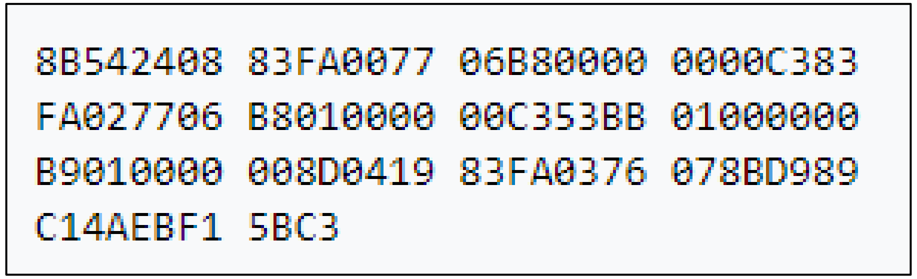
- Arithmetic and Logical Calculations

# Computer Programming

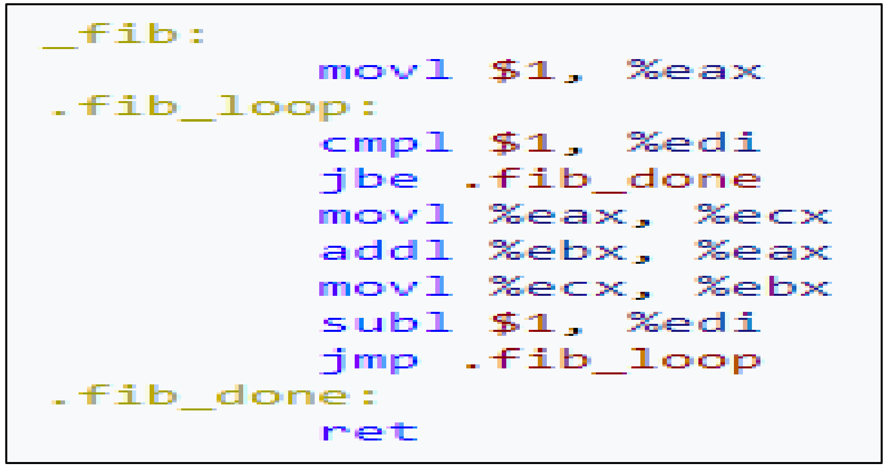
What are the most common elements found in Programming languages ?

1. Variables
2. Statements
3. Control Statements
   A. Decision Making
   B. Looping / Interators
   C. Jump statements
4. Collections / Arrays
5. Functions (Procedure Oriented Programming)
6. Object Oriented Programming
7. File/Stream Handling

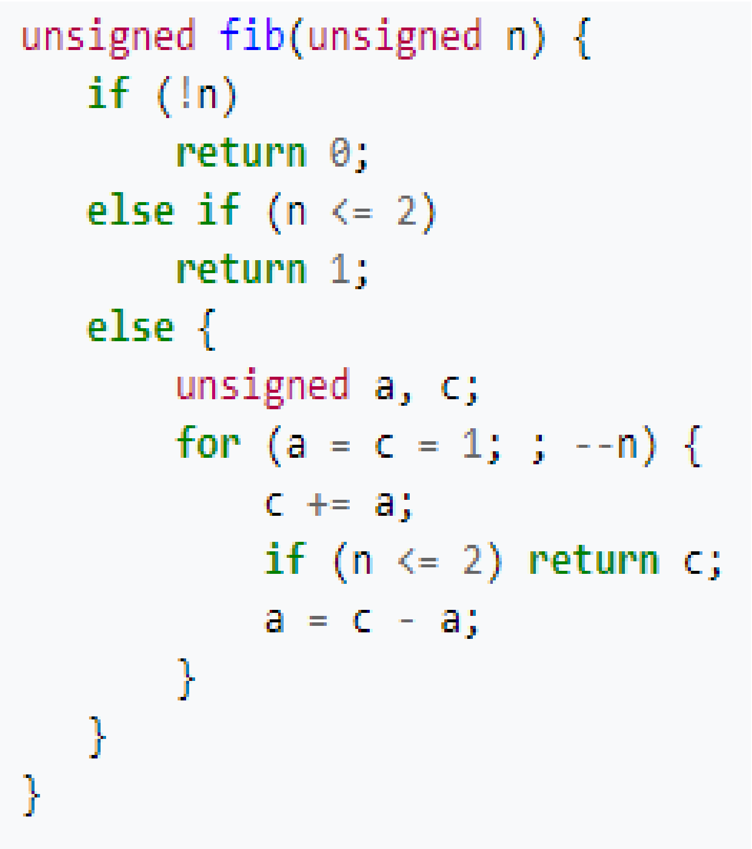
# Programming Languages



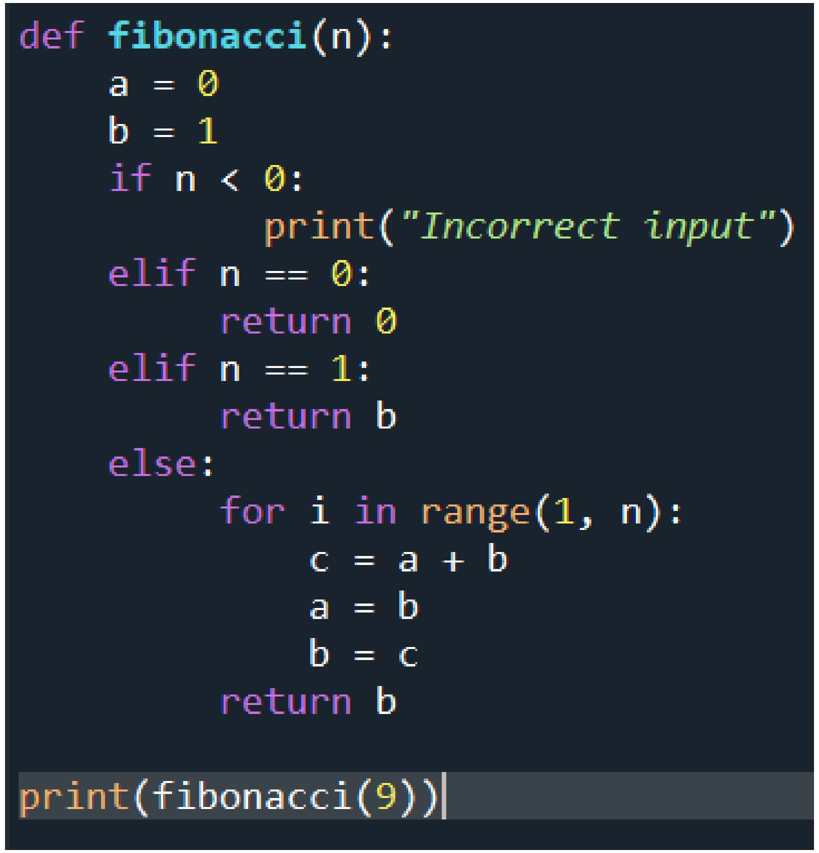
```
8B542408 83FA0077 06B80000 0000C383
FA027706 B8010000 00C353BB 01000000
B9010000 008D0419 83FA0376 078BD989
C14AEBF1 5BC3
```



```
_fib:
        movl $1, %eax
.fib_loop:
        cmpl $1, %edi
        jbe  .fib_done
        movl %eax, %ecx
        addl %ebx, %eax
        movl %ecx, %ebx
        subl $1, %edi
        jmp  .fib_loop
.fib_done:
        ret
```



```c
unsigned fib(unsigned n) {
   if (!n)
       return 0;
   else if (n <= 2)
       return 1;
   else {
       unsigned a, c;
       for (a = c = 1; ; --n) {
           c += a;
           if (n <= 2) return c;
           a = c - a;
       }
   }
}
```



```python
def fibonacci(n):
    a = 0
    b = 1
    if n < 0:
        print("Incorrect input")
    elif n == 0:
        return 0
    elif n == 1:
        return b
    else:
        for i in range(1, n):
            c = a + b
            a = b
            b = c
        return b

print(fibonacci(9))
```

# Programming Languages

- Programming Languages based on Abstraction provided
  - Lower Level Languages
  - Higher Level Language

# Computer Programming

## Who Converts Programming Language to a Language a Machine Understands ?

- Translators like Compiler and Interpreter

# Programming Languages

How Higher-Level Program Executes

- Preprocessor
- Converters (Compiler / Interpreter)
- Loaders (In OS)

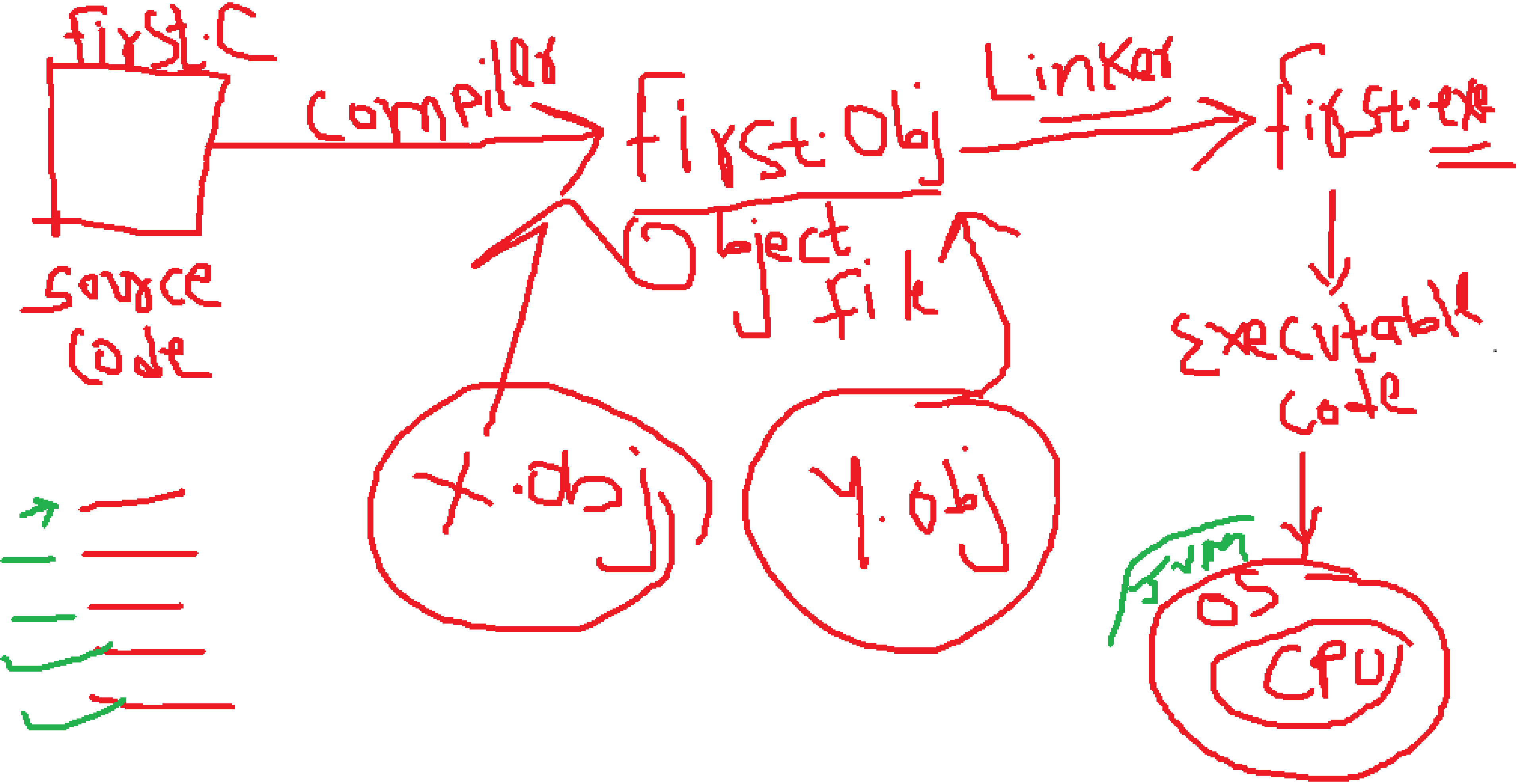
# Programming Languages

| Compiled | Interpreted |
|---|---|
| • Compiler takes entire program as input and converts to machine code | • Takes line by line as input and convert to equivalent machine instructions |
| • Source code converted to machine language form BEFORE execution | • Conversion happens at runtime |
| • Machine specific executable file (.exe) file is generated | • No .exe file |
| • Source code not needed every time to run the program | • Source code needed each time to run |
| • Generally, compiled programs are efficient and execute faster | • More overhead at runtime (syntax checking, linking as well as translation to machine code) |
| • C, C++, Java, C#, Fortran, COBOL | • Python, Java (Compiled and Interpreted), JavaScript, Perl, PHP |

# How Program Executes

- C/C++
- Java
- C#
- Python

# How C Program Executes



# Programming Paradigms

- It is a style, or “way,” of program design.
- It is an approach to solve problem using some programming language.
- Different programming languages follow different approach/style of program design and development
- Some popular and important paradigms are:
  - Structured Programming
  - Procedural Programming / Modular programming
  - Object Oriented Programming

# Structured Programming

Structured programming is a programming paradigm aimed at improving

- ❑ the clarity,
- ❑ quality, and
- ❑ development time of a computer program

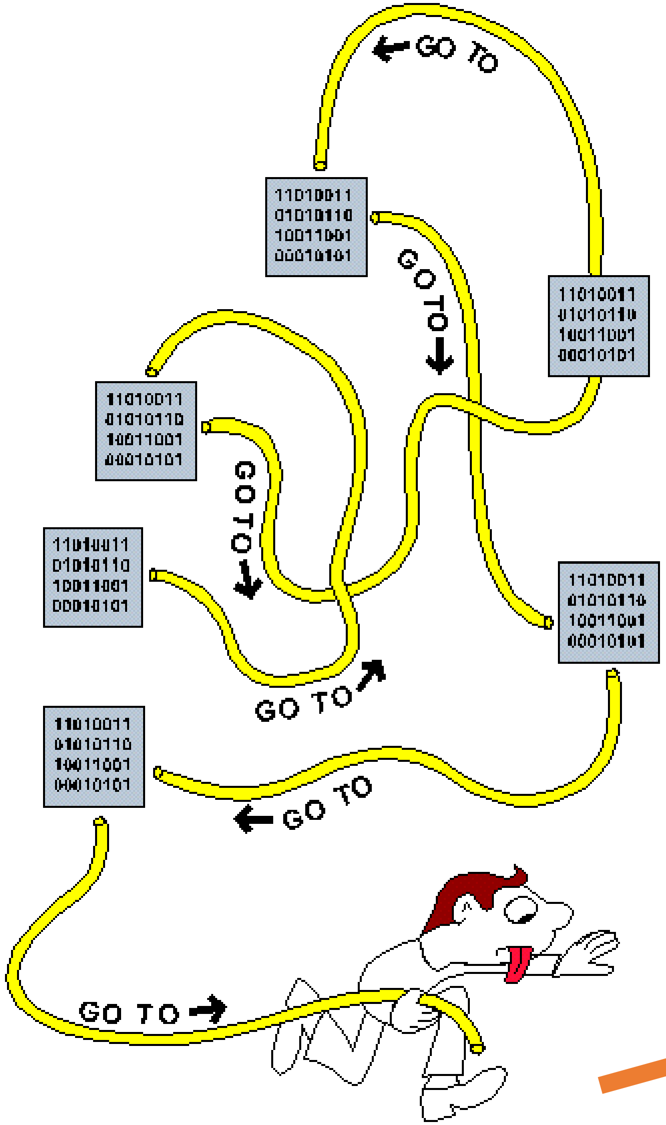
by making **extensive use of the structured control flow** constructs of selection (**if/then/else**) and repetition (**while and for**), block structures, and subroutines.

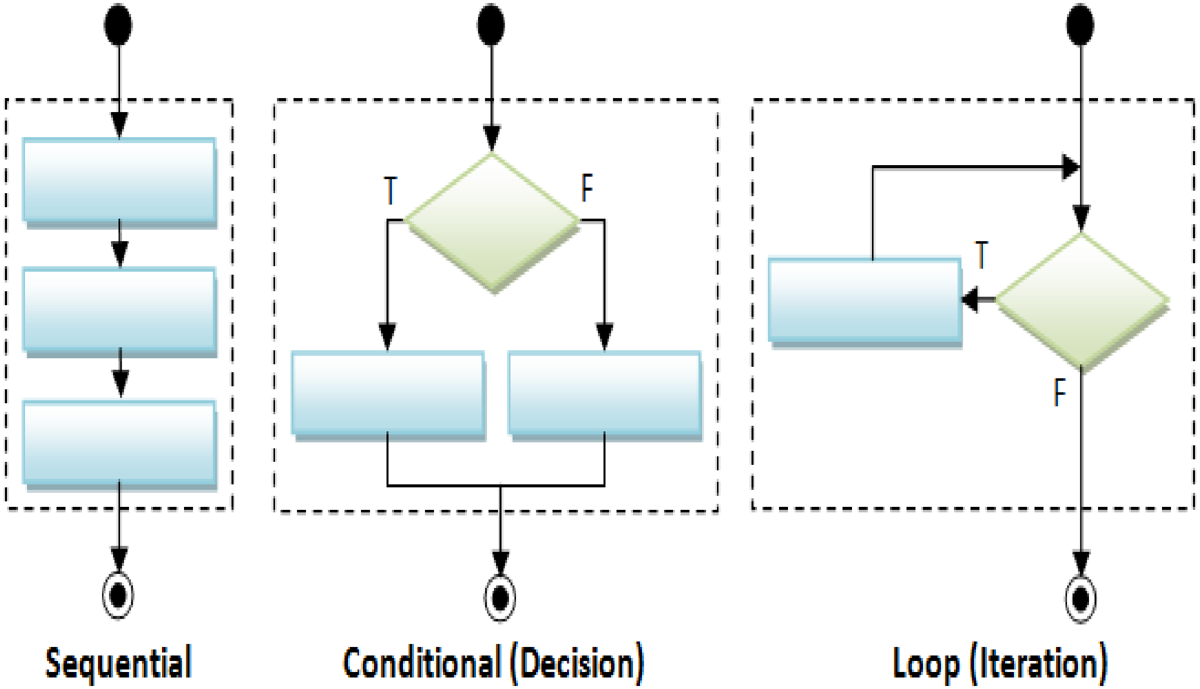
# Structured Programming

- ❑ Program control flow is defined using 3 structures:
  - ✓ Sequences – perform sequence of actions (top-down)
  - ✓ Decisions – perform selection between alternative actions
  - ✓ Loops – perform repetition of same actions

- ❑ **DOES NOT ALLOW to use the jump statement** (GOTO) to transfer program control from one line to another
- ❑ Why ?
- ❑ Use of GOTO increases the complexity of code and program becomes harder to maintain/modify *(results in spaghetti code)*

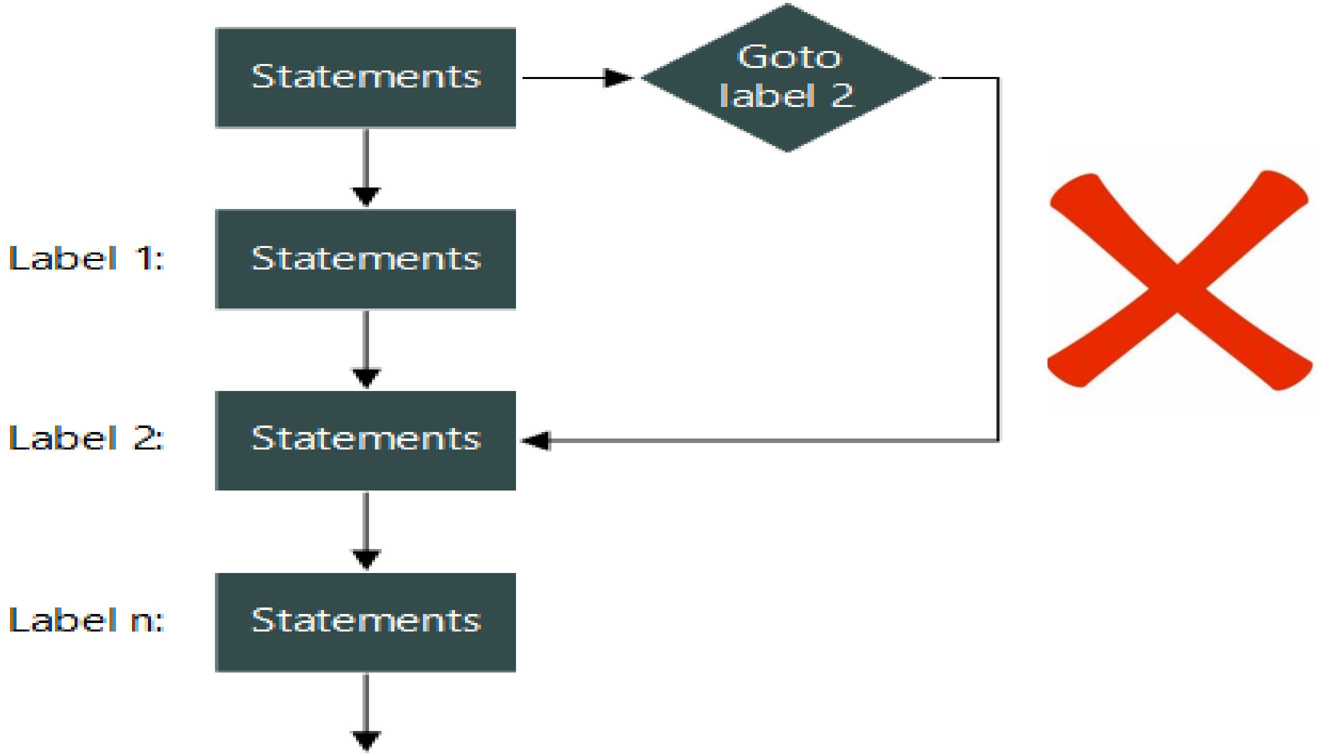
# Structured Programming





# Structured Programming

- Dijkstra, E. W., "*Go To Considered Harmful*," Communications of the ACM, March 1968
- Any program construction could be created more simply with the sequence, repetition and decision constructions WITHOUT USING GOTO

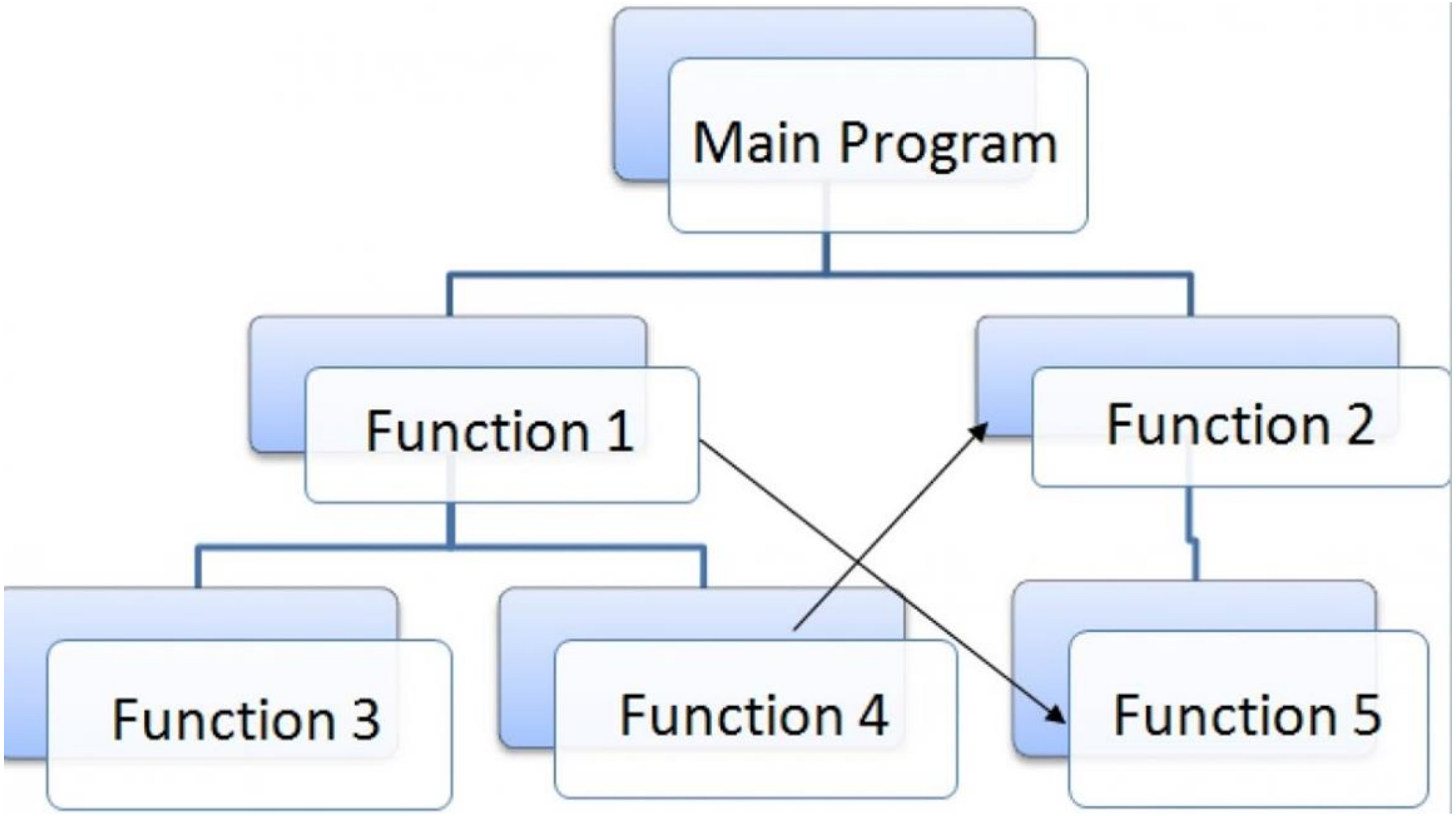


# Procedure Oriented Programming

- Whole program is divided into collection of functions/procedures/modules
- Fundamental building blocks of a program are functions
- Function contains a set of statements to perform a task.
- Provides ability to reuse the code using functions.
- E.g. - C: developed by Dennis Ritchie and Ken Thompson

```
int sum(int num1, int num2) {
   int num3;
   num3 = num1 + num2;
   return (num3);
}
```

# Procedure Oriented Programming

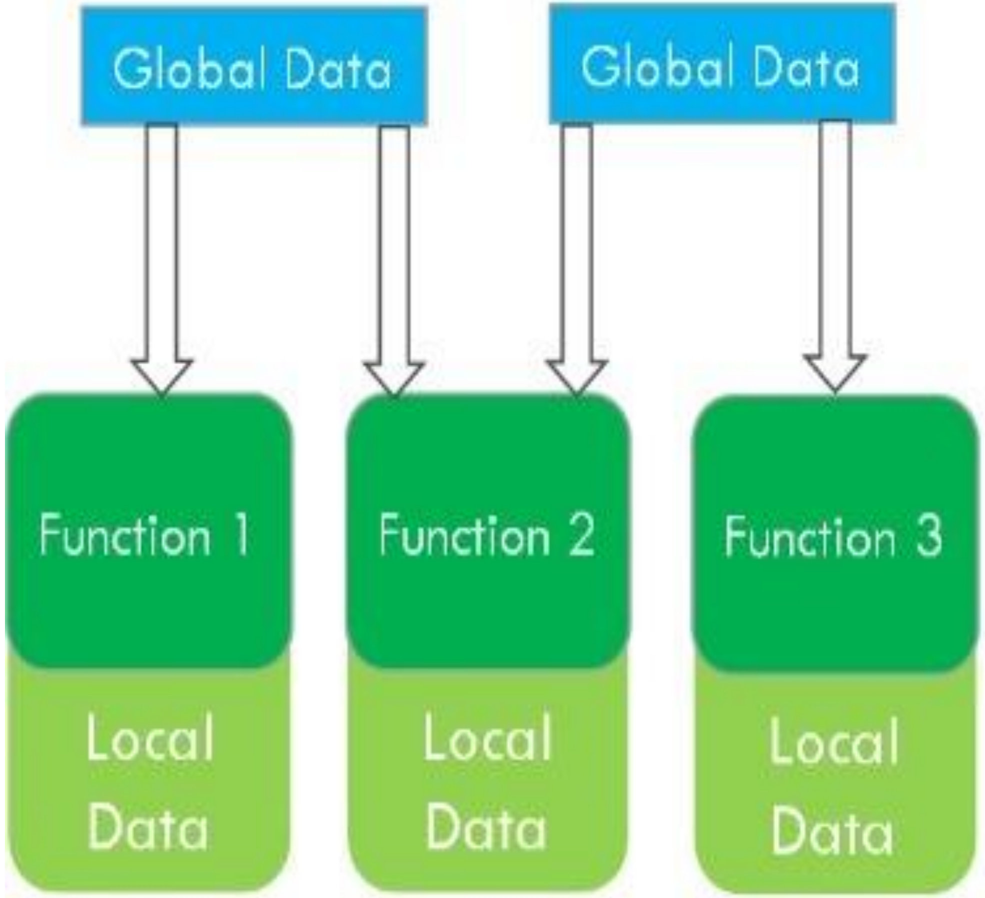


# Object Oriented Programming

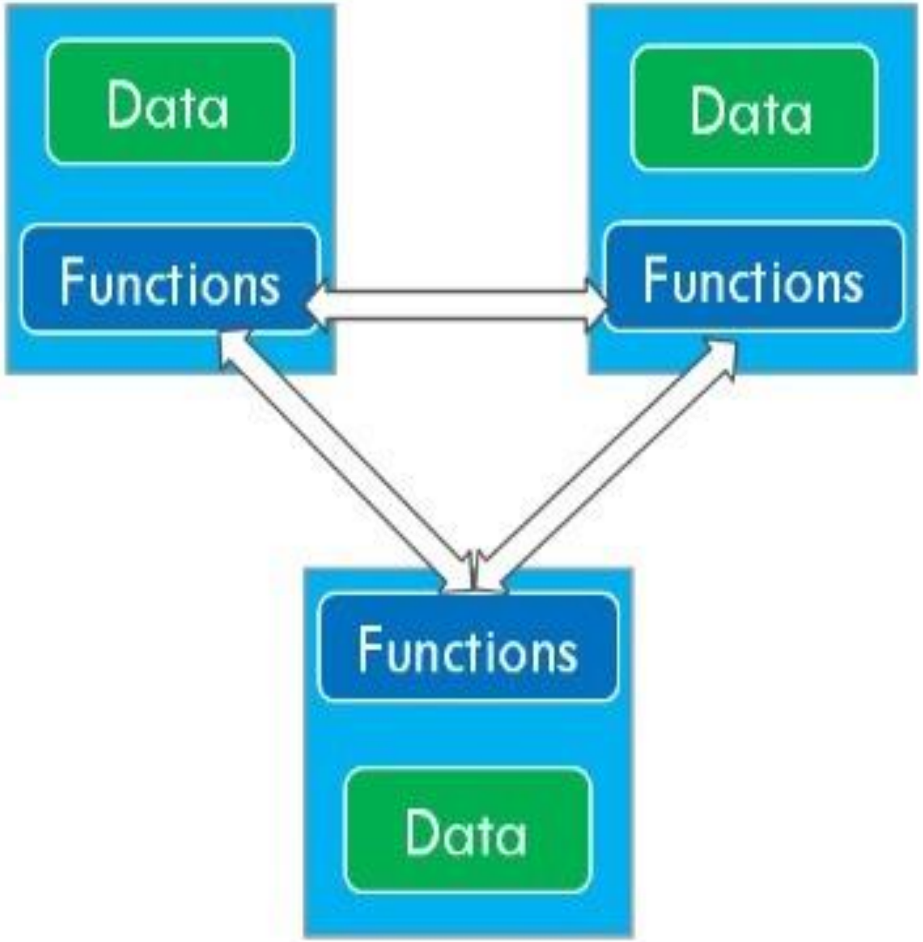
- Design and develop programs around Classes/objects which represent real world entities like Student, Customer, Employee, BankAccount etc.
- Object is the fundamental building block of program
- Object – encapsulates data and functions together into a single unit.
- Advantages - data hiding, code reusability, high level/less complexity and easier to modify/maintain the code, Easier to map real world entities into program

# Procedure Vs Object Oriented Programming

Procedural Oriented Programming



Object Oriented Programming



# References

- B.A. Forouzan And R.F. Gilberg: Computer Science,A Structured Programming Approach Using C, Third Edition, Thomson.
- E. Balaguruswamy, “Programming in ANSI C”, Eighth Edition, Tata Mcgraw Hill Publishing Pvt. Ltd.