LOVELY PROFESSIONAL UNIVERSITY

# CSE109
# Computer Programing

## Lecture #0

The kick start session

# Course Details

- LTP – 2 0 2
- Credits- 3

# Recommended Books

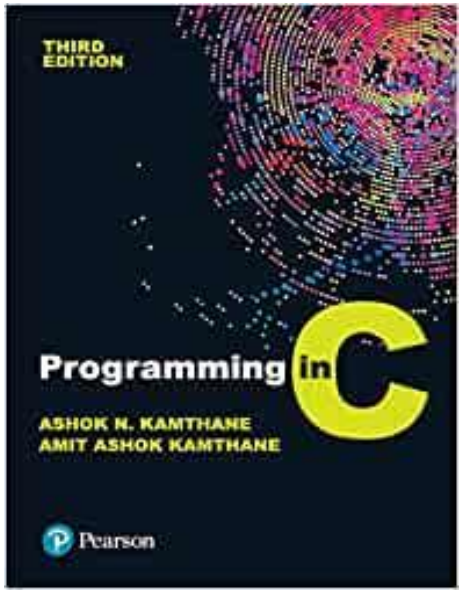
**Text Book**

- "PROGRAMMING IN C"

by

ASHOK N. KAMTHANE

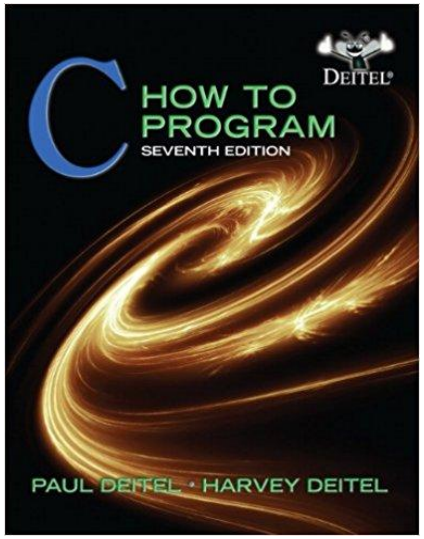
PEARSON, 3rd Edition



# Reference Books

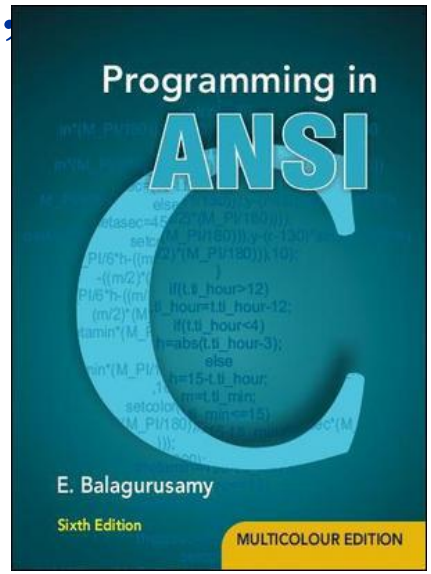
- "C HOW TO PROGRAM" by

PAUL DEITEL AND
HARVEY DEITEL
PHI(Prentice Hall India)




- "PROGRAMMING IN ANSI C"

By E. BALAGURUSAMY
McGraw Hill Education



# Evaluation

1. First two CA's as a written tests, where students have to analyze the problem statements and write relevant programs/debug them.
2. Third CA: Code based third party evaluation

# Academic Task

| Component | Week |
| --- | --- |
| 1. CA1 | 5th |
| 2. CA2 | 9th |
| 3. CA3 | 11th |

Total Weeks: 14 ( 7 Before MTE and 7 After MTE)

# Mode of Conduct

- **BYOD(Bring your own device)**
- **Note: Laptops are mandatory for program implementation.**

# PO/PSO

•PO1:: Critical Thinking: Ability to apply the knowledge of mathematics, science, engineering and technology. Understand in detail, analyze, formulate and solve the issues pertaining to the application of technologies in a range of industrial settings

•PO2:: Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

•PO3:: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

# PO/PSO

•PO4:: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

•PO5:: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

•PO6:: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

# PO/PSO

•PO7:: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

•PO8:: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

•PO9:: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

•PO10:: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

# PO/PSO

•PO11:: Project management and finance: Demonstrate knowledge and understanding of the engineering, management principles and apply the same to one's own work, as a member or a leader in a team, manage projects efficiently in respective disciplines and multidisciplinary environments after consideration of economic and financial factors.

•PO12:: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

# PO/PSO

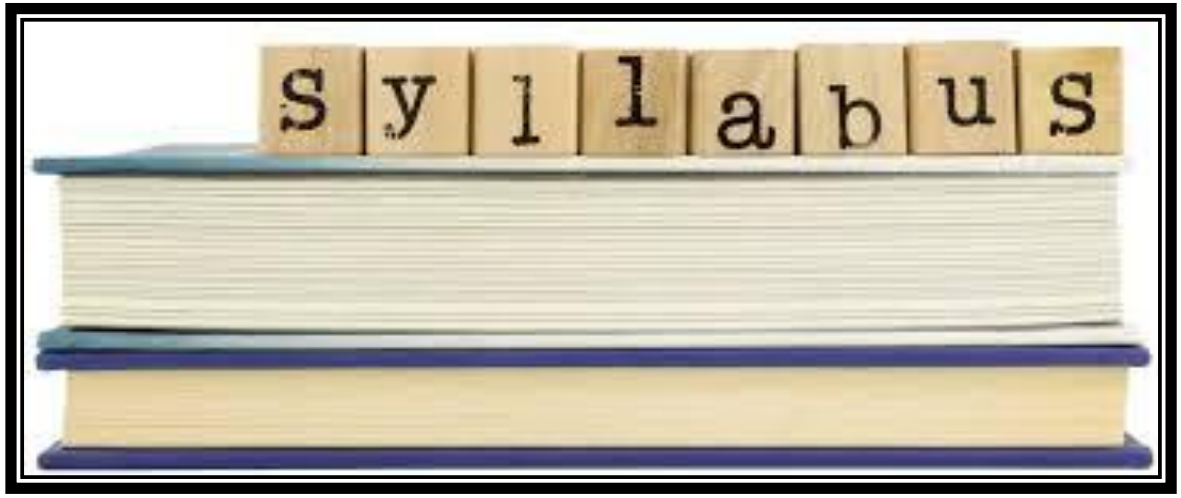
•PSO1:: Demonstrate knowledge and competence in the application of circuit analysis and design in the areas of analog/digital electronics, microprocessor/microcontroller, control systems.

•PSO2:: Demonstrate knowledge and competence in the design, application and analysis of communication systems including wireless, cellular and mobile communication.

# Course Outcomes

- CO1 :: discuss the various approaches towards solving a particular problem using the C language constructs
- CO2 :: write programs to solve different problems using C constructs irrespective of the compilers
- CO3 :: plan the process of code reuse by forming a custom library of one's own functions
- CO4 :: complete the understanding and usage of one of the building blocks of data structures namely pointers
- CO5 :: categorize the theoretical knowledge and insights gained thus far to formulate working code
- CO6 :: validate the underlying logic and formulate code which is capable of passing various test cases

# Syllabus


s y l l a b u s


INSTRUCTION
STRATEGY
LEARN
EDUCATIONAL
COLLEGE
TEACH
INFORMATION
DISCOVER
STUDYING
PLAN
PROJECT
GOAL
SUCCESS
SOLUTION
STUDENT
TEACHER
INTELLIGENCE

# Course Content

- Basics and introduction to C
- Control structures
- Input/output functions
- User defined functions
- Storage classes
- Arrays
- Pointers
- Dynamic Memory Allocation
- Strings
- Derived types

# Course Content

- **Basics and introduction to C**

Components of C character set, discussion on identifiers, keywords and data types. awareness of the basics of C language without which it is not possible to work with C language.

- **Control structures**

if, if else, switch case statements, while loop for, do while loops, at least 2 examples should be discussed for all the constructs. Awareness of which statements to use while dealing with various problems.

- **Input/output functions**

printf, scanf functions along with various format specifiers. Students will learn about the suitability of various input and output statements for handling different types of data

# Course Content

- **User defined functions**

Description about user defined functions, methods of calling a function and function prototypes. You will be able to write customized functions according to the given requirement and will learn modular approach of programming

- **Storage classes**

Lifetime of a variable, Visibility of a variable, Various storage classes such as automatic, external, static and register, example in context to function.

- **Arrays**

introduction to arrays, declaration, initialization of arrays. passing array as a function argument, few sample programs of passing arrays to functions,Insertion and deletion from different positions from array

# Course Content

- **Pointers**

Need of pointers, declaring and initialization of pointer variables, various operators such as address operators, indirection operator, types of pointers including void, wild and null pointers.

- **Dynamic Memory Allocation**

Different function used for dynamic memory allocation.

# Course Content

- **Strings**

String basics including the use of character arrays to store and manipulate strings, reading and writing from and to strings.

- **Derived types**

Introduction to structures including the need of structures,defining and assigning values to a structure, operations which can be carried out on structure members after accessing them, few examples on structures,Creating a pointer to structure,Creating a structure with in a structure

# Cohort Description

Software Engineer/ Software Developer

This cohort imparts skills to:

- Analyze the user requirements
- Develop the software applications
- Maintain and update software applications
- Team work for projects

# Course Enrichment

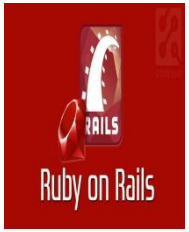
1. Github repository creation
2. Alumni interaction/placed students
3. Mock interview
4. Profile creation on online platforms
5. Sensitization of students about the importance of ranking on online platforms

# Lets take you close to the reality

- All of us use Computers or Laptops for different purposes.
- Could you tell me which system software is most required to our system in work mode?????

# WHY C??????????????

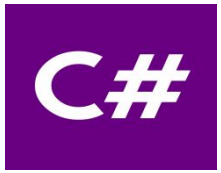






- If we have number of powerful programming languages available with us then why c??????







# The hitch.....

Some burning questions in mind.....

- C is a very old language. Why are we still studying this language??????
- Now, we have very powerful languages with us then, why c??
- There is no scope of this language in industry

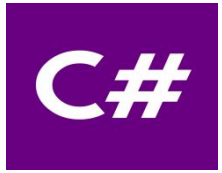

# Let’s Explore more

- Could you tell me which programming language is used in writing all these operating system??????


- Latest version of Microsoft Windows i.e. Windows11 is still being written in C Language


# Contd.....

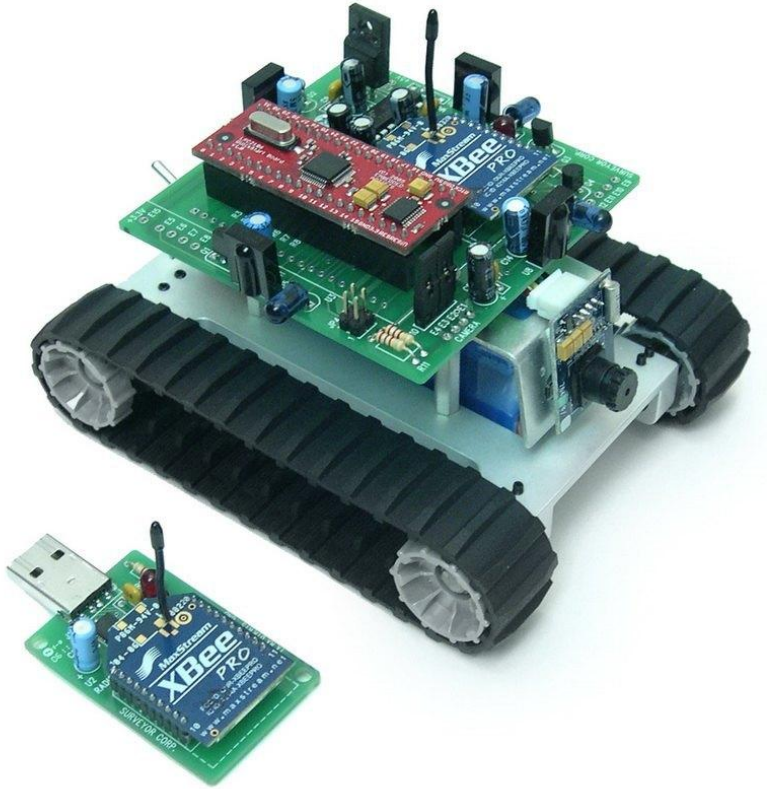
- Device drivers are also written in C language.
- All these modern programming languages are influenced by C language




- Compilers for Python and PHP language are also written in C language

# Contd.....

- Embedded systems are also developed with the




# Contd.....

- Git 

- Microsoft Excel 

- Oracle Database 

- MySql 

- Linux 

- Unix 

- Android 

- Google 

# MNCs

*Top rated Companies which has a dearth of C programmers*

# Here are the Answers of Questions

- C is very a old language still, why do we study C language??


- Now, we have very powerful language with us then why c??


- There is no scope of this language in industry

# Another Burning questions:

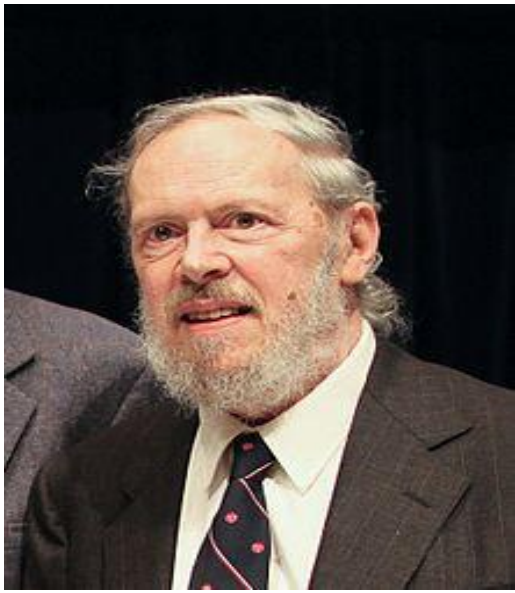
How C is better than Python ?

# History of C

- Guys Can you make a sentence with the word '***Necessity***'

  "Necessity is the mother of invention"

- Dennis Ritchie and Ken Thompson were working on developing a new operating system i.e UNIX
- But the programming language they were using was not providing them the portability feature
- So Dennis Ritchie developed new language i.e C



# History continued...

| Summary - | | |
|---|---|---|
| 1 | B Language Developed By | Ken Thompson |
| 2 | Operating System Developed in C | UNIX |
| 3 | Developed at | AT & T Bell Laboratory |
| 4 | Creator of Traditional C | Dennis Ritchie |
| 5 | Year | **1972** |

# Why “C” name was given???

- Many of C’s principles and ideas were derived from the earlier language B. (Ken Thompson was the developer of B Language.)
- BCPL and CPL are the earlier ancestors of B Language
- CPL is Combined Programming Language. In 1967, BCPL Language ( Basic CPL ) was created as a scaled down version of CPL
- As many of the **features were derived from “B” Language thats why it was named as “C”**.
- After 7-8 years C++ came into existence which was first example of object oriented programming .

# Evolution of C...


| | |
|---|---|
| **2011** | **C11** |
| **2018** | **C17(Current)** |
| **Upcoming** | **C2x** |

# Language Developers

| | |
|---|---|
| Algol | • International Group |
| BCPL | • Martin Richards |
| B | • Ken Thomson |
| Traditional C | • Dennis Ritchie |
| K&R C | • kernighan & Ritchie |
| ANSI C | • ANSI Commitee |
| ANSI/ISO C | • ISO Commitee |
| C99 | • Standerd Commitee |

# Features of C Language

- Low Level Language Support
- Program Portability
- Powerful and Feature rich
- High Level Features
- Modular Programming
- Also known as middle level language

# Applications of C

- Used for creating computer applications
- Used in writing embedded software
- Development of Simulators
- Used for creating compilers
- Used to implement different  operating system operations
- UNIX kernel is completely developed in C language

# Course Contents

## Before MTE

- ✔ Data Types & Operators
- ✔ Control Structures
- ✔ User Defined Functions
- ✔ Storage Classes

## After MTE

- ✔ Arrays and Strings
- ✔ Pointers
- ✔ Dynamic Memory Allocation
- ✔ Derived Data Types- Structures and Union

**Who is father of C Language?**

A. Bjarne Stroustrup
B. James A. Gosling
C. Dennis Ritchie
D. Dr. E.F. Codd

**Which of the following statement is correct about the C language?**

1. The C language is a binary language with some extra features.
2. The C language is a high-level language with some low features.
3. The C language is a mid-level language with some high features.
4. The C language is a low-level language.

**C Language developed at _____?**

A. AT & T's Bell Laboratories of USA in 1972
B. AT & T's Bell Laboratories of USA in 1970
C. Sun Microsystems in 1973
D. Cambridge University in 1972

# Now its your turn to tell:

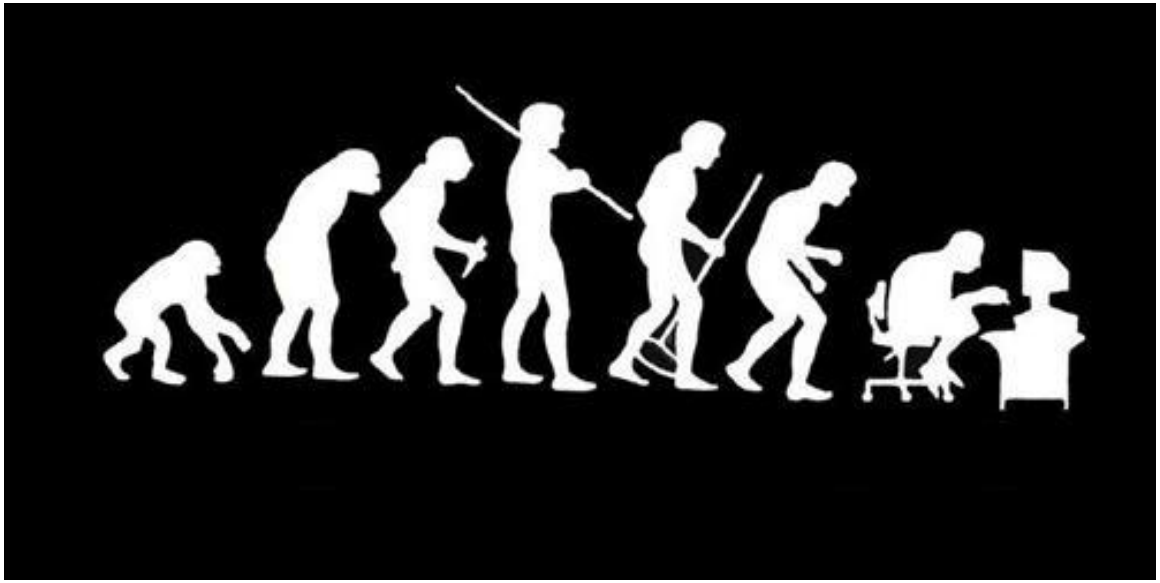
- **Why are we learning C language?**
- **What would we do with it, we are engineers?**

# Let us re-invent ourselves

To begin with basics…

Let us go to basics.

Let us begin from toddling to learn to walk

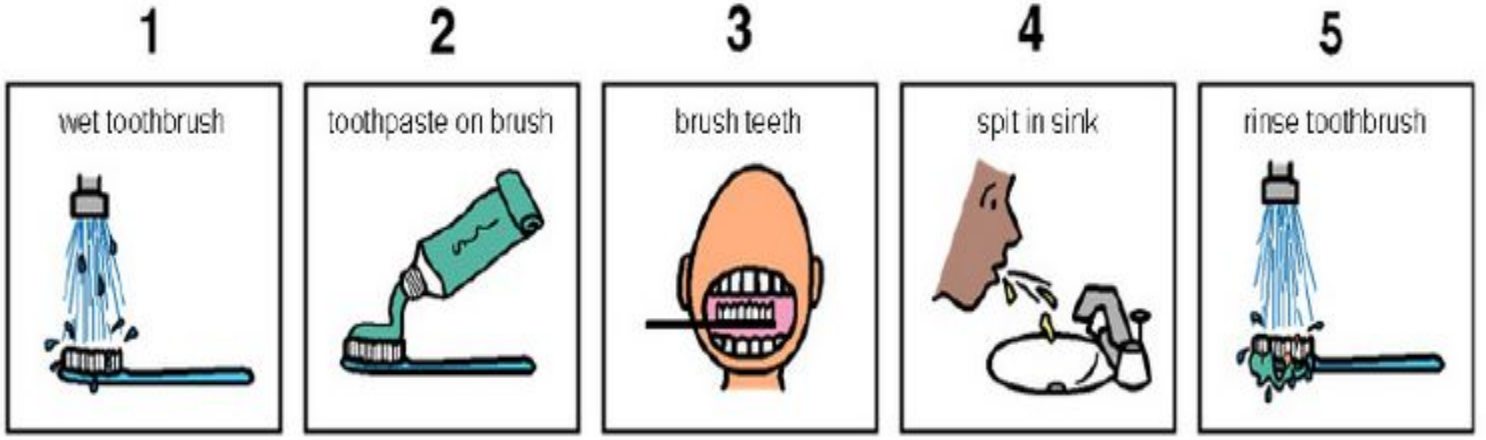


**Get ready to be**

# Daily routine

Let us look around our daily routine…

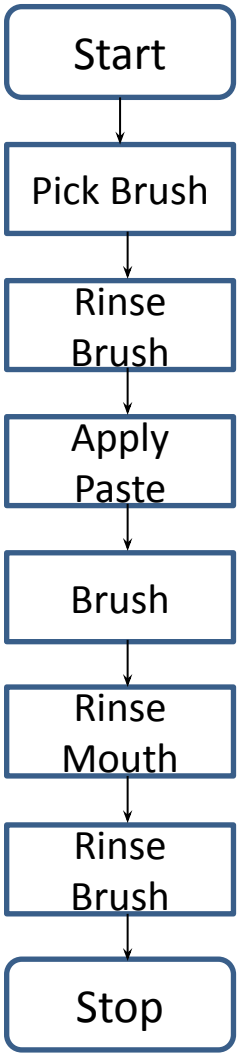
- **Let us see where all we do programming everyday**
- **Simple things we do to start the day**




**So there is ONE program you know which is there in**

# Daily routine

- **There is a set procedure**
- **Each step is defined**
- **The occurrence is ordered**
- **Jump is NOT permitted**
- **A step cannot be skipped**




# Daily routine

- **Let us explore more as the day goes by…**

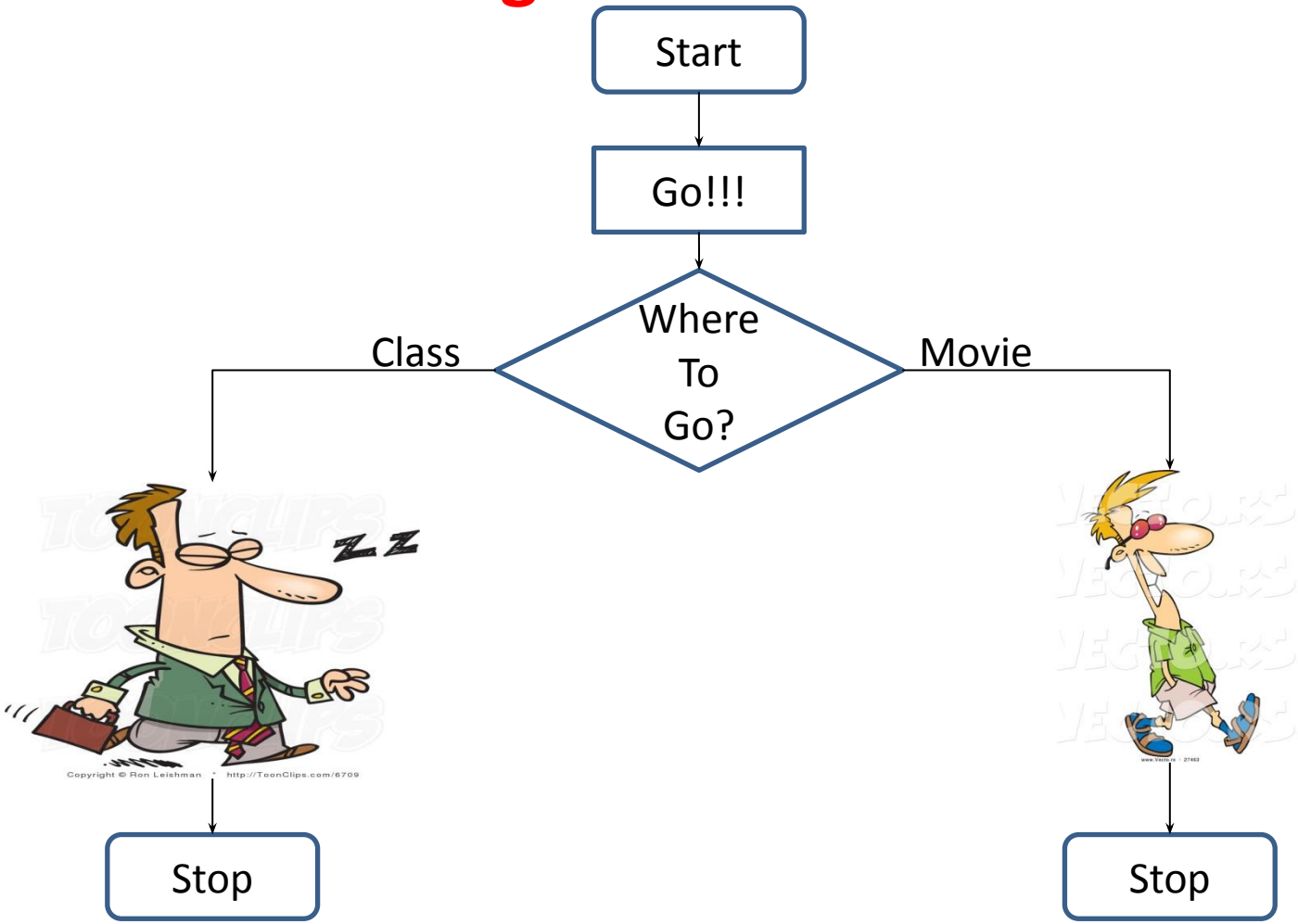

**Going for a morning 0900 AM Class**

**Going for a movie at 0900 AM**

**It is all about <u>WHICH</u> program is loaded <u>WHEN</u>**

# Daily routine

- **The flow changes**



# Daily routine

- **Yet another example but more complex**


Vision Sensor

Smell Sensor

Radiation Sensor

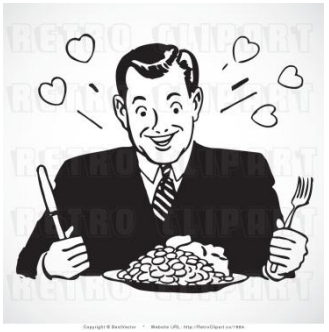
Touch Sensor

If all sensors ✓ = SIP

If even one is ✗ = WAIT!!!

# So what does this mean?

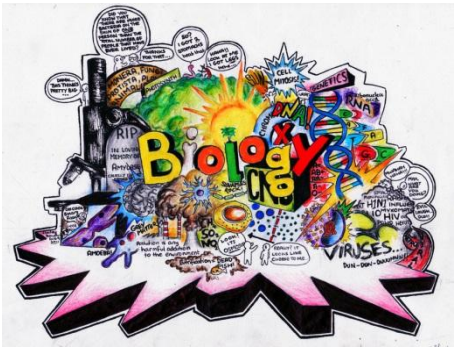

- **Take ANY activity of the day…**
- **It will have a set procedure**
- **It has to be done in a designate way**
- **If not done the specified way will yield wrong results**
- **Success in doing it depends on how closer one is to the prescribed method**



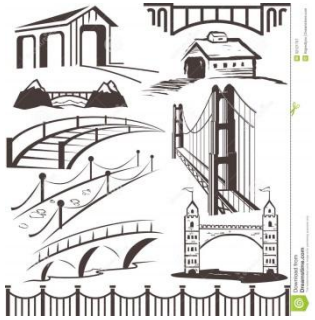
- **This clearly shows that everything has a**

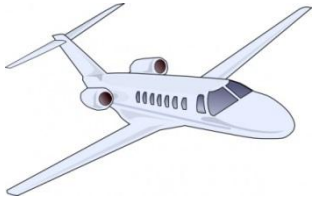
**Logic**

# Logic, logic and logic


**Logic**





# What next?

- If there is a logic in anything and everything
- **There has to be ways to represent logic**
- **There has to be modes to modify and re-represent logic**
- **There should be methodology to implement and re-design logic**
- **And for all this…**

# What next?

- **There has to be logic machine to assimilate, understand, solve, store, retrieve and represent logic**


- **There has to be a <u>LANGUAGE</u> to communicate with the logic machine**

**Otherwise…**

.

# Diving deeper…

# Some Practical Applications

- C is one of the oldest and most fundamental programming languages, and it is extensively used all over the world.
- Some applications of the C programming language include:
- Operating System: The C programming language was created with the intention of writing UNIX operating systems. It was used to write the Unix kernel, Microsoft Windows utilities and operating system apps, and a large portion of the Android operating system.
- 3D Movies: Applications written in C and C++ are commonly used to make 3D videos, because they handle a large quantity of data and do many computations per second, these apps must be extremely efficient and quick. The less time it takes for designers and animators to create movie shots, the more money the corporation saves.
- Intermediate Language: C is occasionally used by implementations of other languages as an intermediate language. This method can be used for portability or convenience, as it eliminates the need for machine-specific code generators by using C as an intermediate language.
- Play Important Role in Development of New Programming Language: The program written in C is easy and quick to execute. As a consequence, the C programming language has resulted in the creation of many other languages. C++ (also known as C with classes), C#, Python, Java, JavaScript, Perl, PHP, Verilog, D, Limbo, and the Unix C shell are examples of these languages.
- Embedded Systems: The C programming language is the recommended language for creating embedded system drivers and applications. The availability of machine-level hardware APIs, as well as the presence of C compilers, dynamic memory allocation, and deterministic resource consumption, make this language the most popular.

# Programming for Engineers…

# Pedagogical Innovations

1. Compiler based live demonstration of the concepts
2. Real life examples/scenarios covered for problem solving through programming
3. Implementation of tasks on online platforms
4. Case studies given for implementation
5. Industry level situation based detailed questions and MCQs are given.
6. Practice question given on third party platform such as hackerrank.
7. Awareness and motivating for Hackathons/online competitions

# Relevant Resources

1. https://www.geeksforgeeks.org/c-programming-language/
2. https://www.programiz.com/c-programming
3. https://www.tutorialspoint.com/cprogramming/index.htm
4. https://freevideolectures.com/course/2519/c-programming-and-data-structures
5. https://www.youtube.com/playlist?list=PLBlnK6fEyqRggZZgYpPMUxdY1CYkZtARR

# Get Set Go!!!

**Gear up**
**Fasten your seat belts**


**Explore your logical abilities**

**Build your packages for future support**

Next Class: Basics and introduction to C