CAP444

OBJECT ORIENTED PROGRAMMING USING C++

Lecture #0
The kick start session

Presented By
Kumar Vishal
Assistant Professor
SCA, LPU

Course details

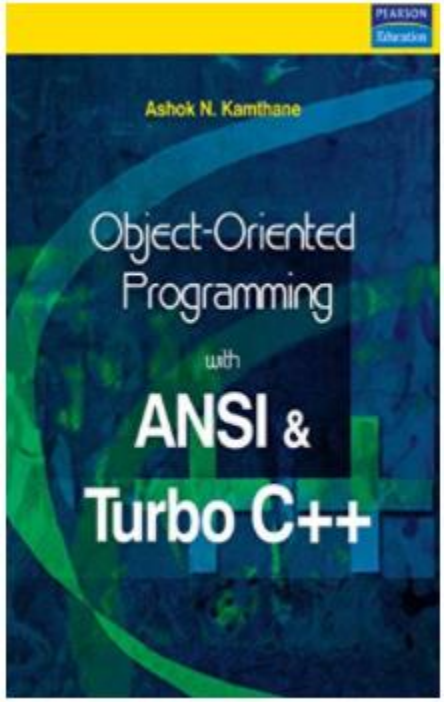
- LTP – 3 0 0 [Three lectures/week]
- **Text Book**

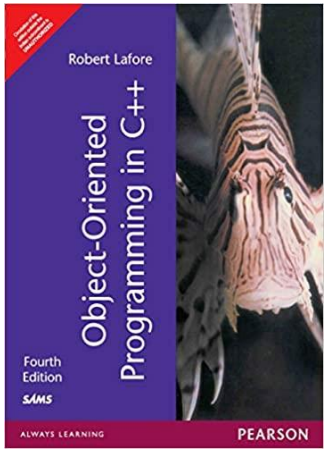
OBJECT ORIENTED PROGRAMMING WITH ANSI & TRUBO C++ by ASHOK N. KAMTHANE, PERASON EDUCATION

- **Reference Books:**

OBJECT ORIENTED PROGRAMMING IN C++

BY ROBERT LAFORE, GALGOTIA PUBLICATIONS





Course Assessment Model

Marks break up	
Attendance	5
Continuous Assessment(2 out of 3)	25
MTE	20
ETE	50
Total	100

The hitch…

The three BURNING questions in mind…

- Why are we learning C++ language?
- What would we do with it?
- What will be the course outcome?

Course Outcomes:

CO1 :: define the various concepts of object oriented programming

CO2 :: understand the working with files and streams

CO3 :: practice the generic programming to increase the efficiency of code

CO4 :: analyze the unexpected situations and manage them using exception handling mechanism

Programme Outcomes

1. Computational Knowledge:

Apply knowledge of computing fundamentals, computing specialization, mathematics, and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.

2. Problem Analysis:

Identify, formulate, research literature, and solve complex computing problems reaching substantiated conclusions using fundamental principles of mathematics, computing sciences, and relevant domain disciplines.

3. Design /Development of Solutions:

Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

Programme Outcomes

4. Conduct investigations of complex Computing 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.

5. Modern Tool Usage:

Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.

6. Professional Ethics:

Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practices.

7. Life-long Learning:

Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.

Programme Outcomes

8. Project management and finance:

Demonstrate knowledge and understanding of the computing and management principles and apply these to one ’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

9. Communication Efficiency:

Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

10. Societal and Environmental Concern:

Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practices.

11. Individual and Team Work:

Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

12. Innovation and Entrepreneurship

Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

Mapping of POs with COs

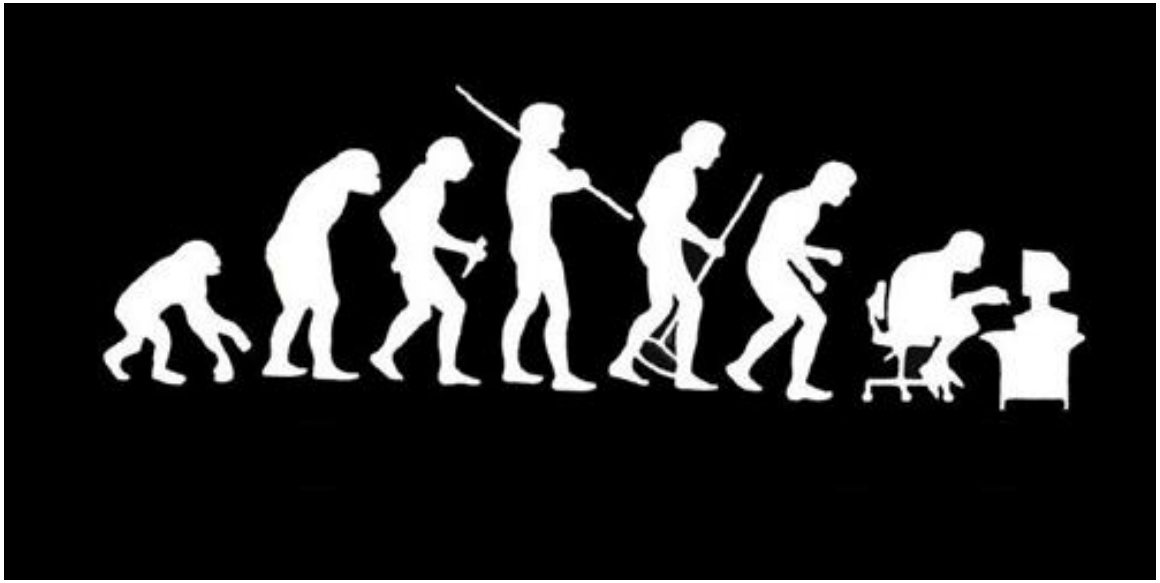
	CO1	CO2	CO3	CO4
PO1	3	3	3	3
PO2				3
PO3	2	2	3	3
PO4				
PO5	2	2	2	2
PO6	1	1	1	1
PO7	3	1	1	1
PO8				
PO9				
PO10				
PO11				
PO12				

Let us re-invent ourselves

To begin with basics…

Let us go to basics.

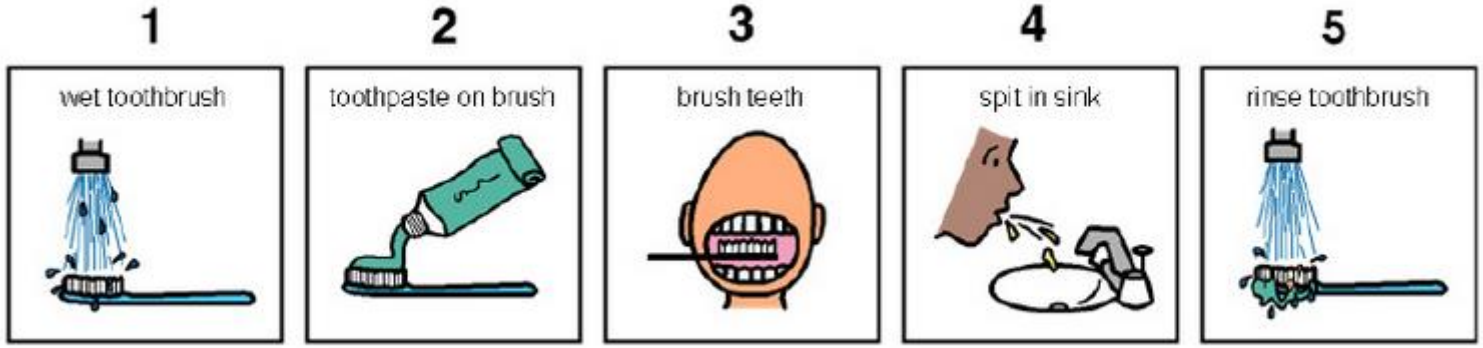
Let us begin from toddling to learn to walk



Get ready to be childish….

Daily routine

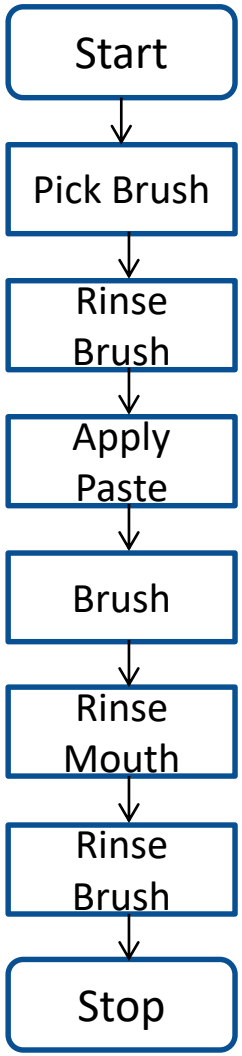
- Let us look around our daily routine…
- Let us see where all we do programming everyday
- Simple things we do to start the day



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

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…

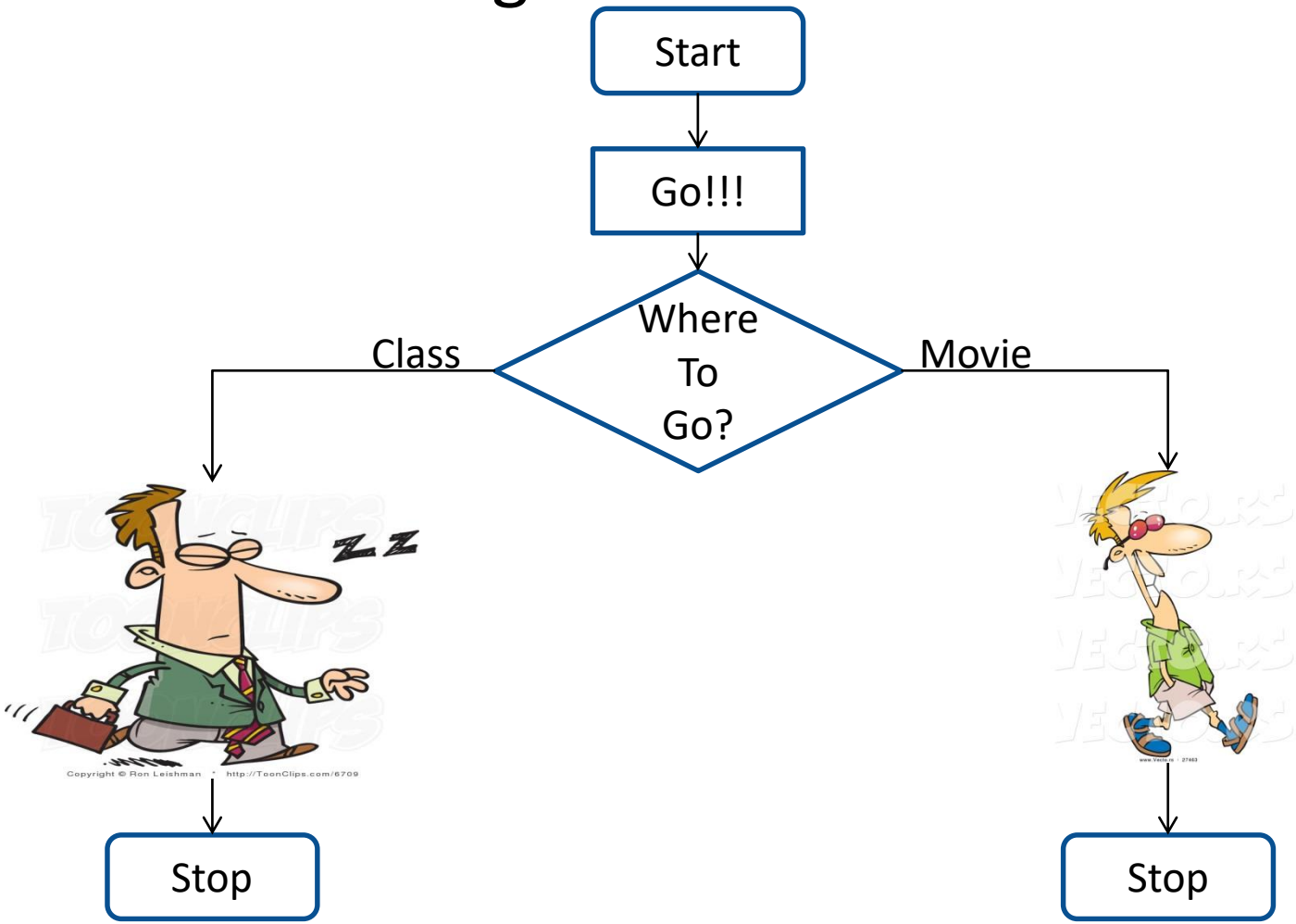
Going for a morning 0900 AM Class

Going for a movie at 0900 AM

It is all about WHICH program is loaded WHEN

Daily routine

- The flow changes



Daily routine

- Yet another example but more complex

Vision Sensor

Smell Sensor

Radiation Sensor

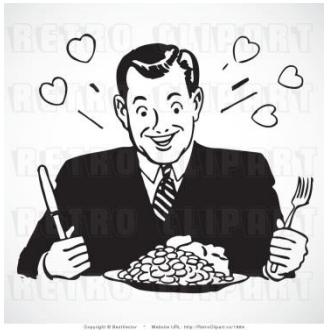
Touch Sensor

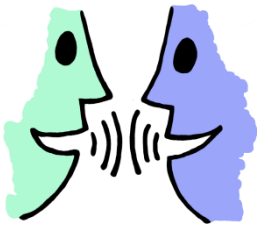
If all sensors = SIP

If even one is = WAIT!!!

So what does this mean?

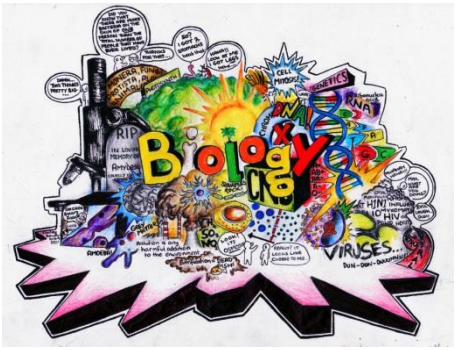
- Take ANY activity of the day…
- It will have a set procedure
- It has to be done in a designated way
- If not done the specified way will yield wrong results.
- Success in doing it depends on how closer one is to the prescribed method.
- This clearly shows that everything has a



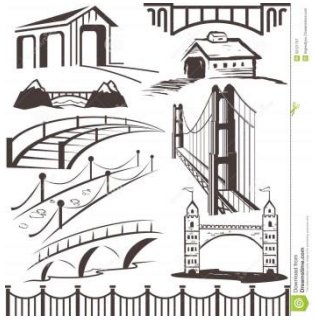


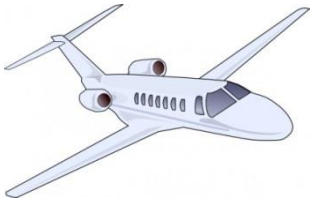
Logic

Logic, logic and logic



Logic





What next?

- If there is 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…

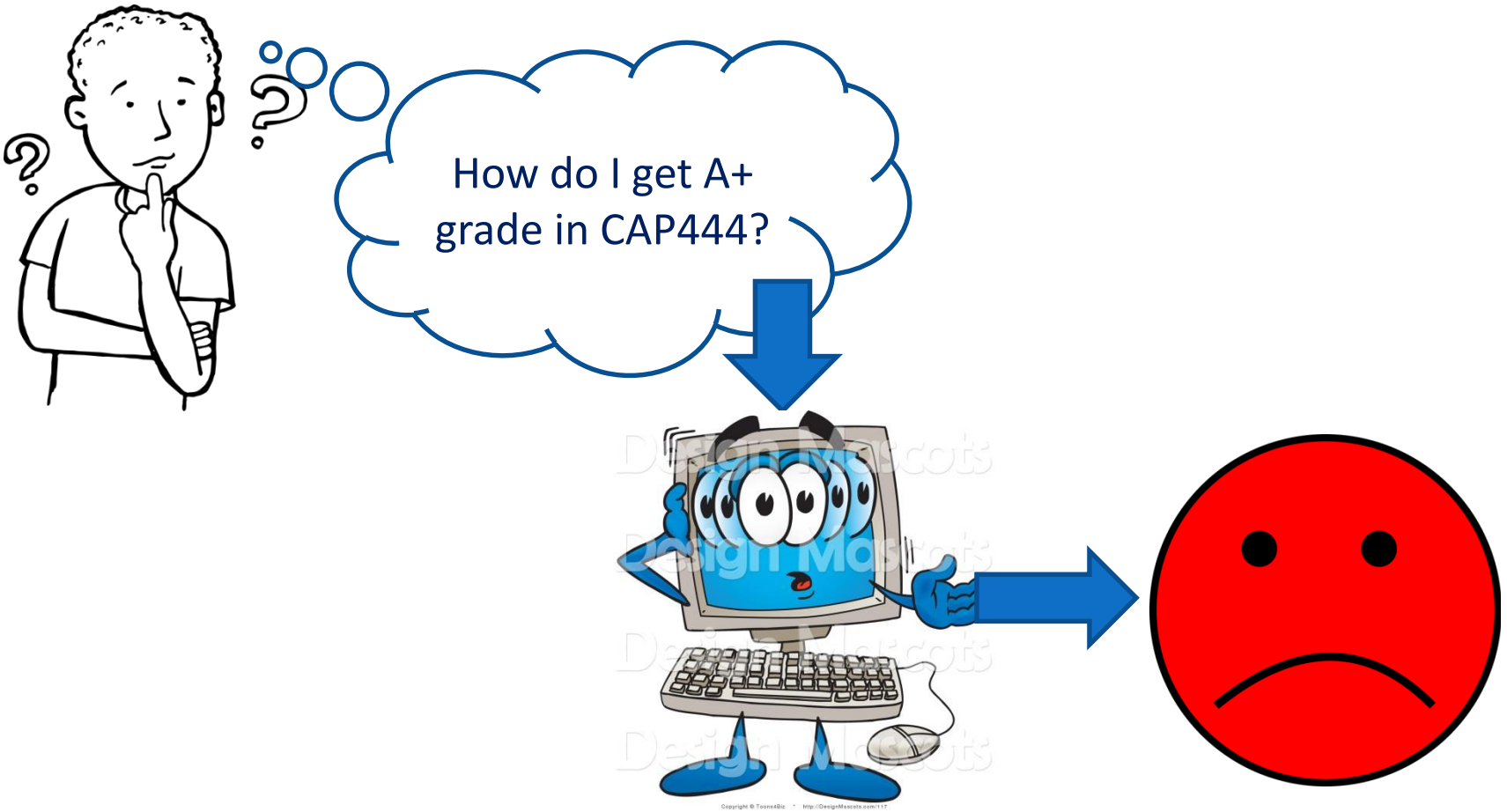
What next?

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

- There has to be a LANGUAGE to communicate with the logic machine

Otherwise….

Diving deeper…



Diving deeper…

How do I get A+ grade in CAP445?

If marks are greater than or equal to 85. A program written in C++ language with **LOGIC**

Course Content

Principle of OOP's
- classes and objects
- the concept of constructors and destructors
- friend Functions

Inheritance and type conversion
- different types of inheritance
- basic to class type conversion
- class type to basic type conversion
- class type to class type conversion

Polymorphism
- unary operator overloading
- binary operator overloading
- abstract classes
- virtual functions and pure virtual functions
- this pointer
- pointer to object

Working with Files and Streams
- different file operations
- the concept of random access in files
- the concept of command line arguments

Generic Programming with Templates
- class and function templates
- function template overloading
- recursion with template function
- macros

Exception handling
- exceptions handling mechanism
- multiple exceptions handling
- exceptions in constructors and destructors

Click Me

What do we need to know?

Unit 1

Tamil

தமிழ்

Punjabi

ਪੰਜਾਬੀ

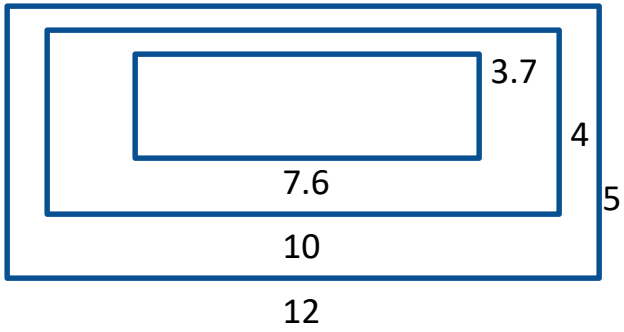
Programming Language

C++

program

What do we need to know?

Unit 1



Principles of OOP's and C++ Basics

- How to do calculations

  Area = Length * Breadth

  Area = 12 * 5

  Area = 10 * 4

  Area = 7.6 * 3.7

What do we need to know?

Unit 1

Classes and Objects

Mobile

Symbian (Keypad Phone)

i-phone

Android phone

What do we need to know?

Unit 1

Constructors and Destructors

Inheritance and its types

father

mother

child

What do we need to know?

Unit 2

Operator Overloading and Type Conversions

One socket board → multiple work

Ice to water

What do we need to know?

Unit 3

Run-time Polymorphism and Virtual Functions

Preparing food according to hotel menu

Compile time)

During competition preparing food

(Run-time)

What do we need to know?

Unit 4

Working with Files and Streams

What do we need to know?

Unit 5

Generic Programming with Templates

interview

Resume Format

What do we need to know?

Unit 6

Exception Handling

Acknowledgements

- The Khan Academy
- EdX
- Coursera
- Cplusplus.com
- www.cppforschool.com
- Learncpp.com
- And Above all…

COMMON SENSE

Next :Principles of OOP's