

# CSE202

## Object Oriented Programming

L:3 T:0 P:2

Credits: 4

### Lecture 0

Welcome!!!!

# Vision

---

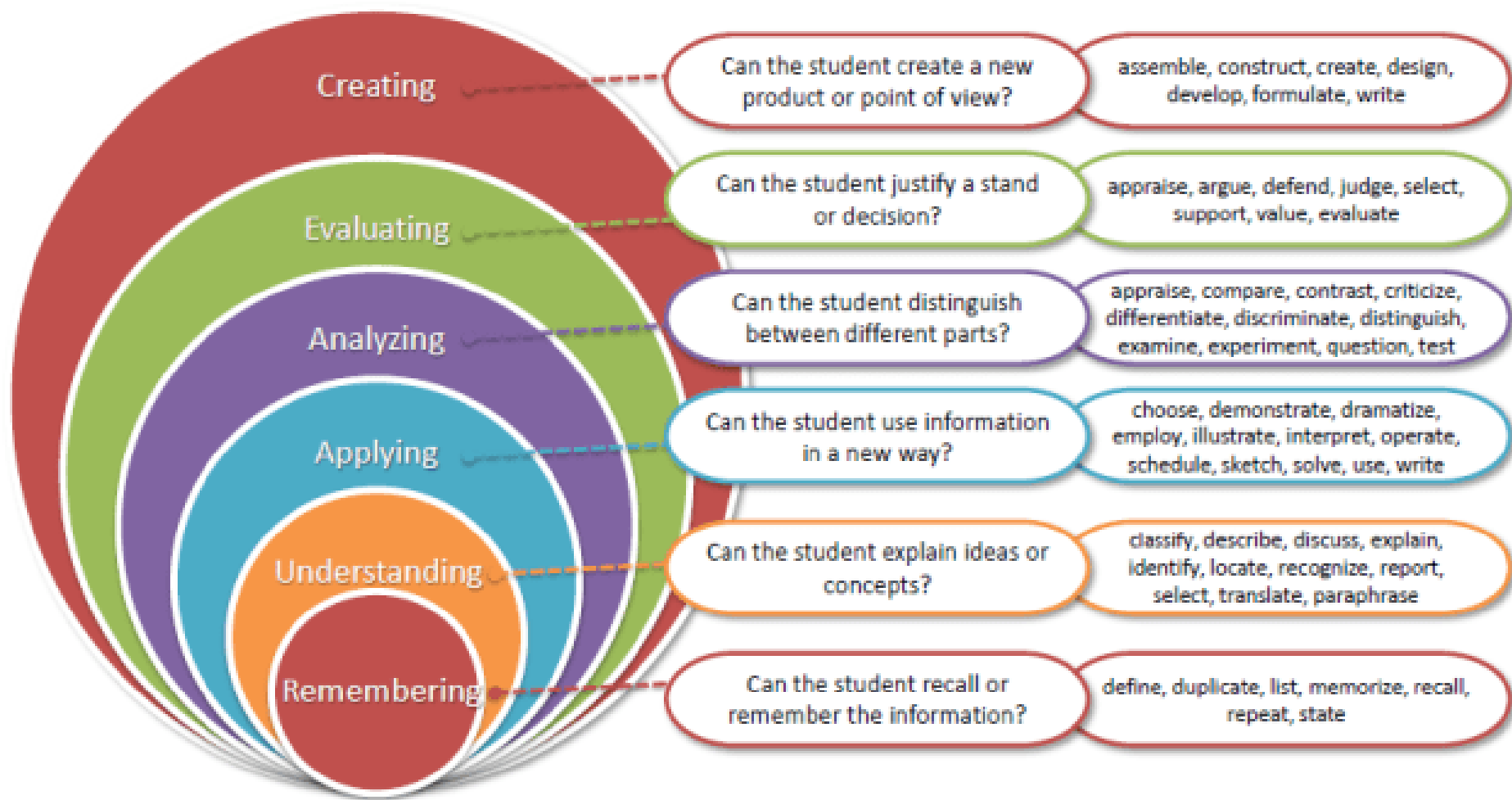
To be a globally recognized school through excellence in teaching, learning, and research for creating Computer Science professionals, leaders, and entrepreneurs of the future contributing to society and industry for sustainable growth.

# Mission

---

- To build computational skills through hands-on and practice-based learning with measurable outcomes.
- To establish a strong connect with industry for in-demand technology driven curriculum.
- To build the infrastructure for meaningful research around societal problems.
- To nurture future leaders through research-infused education and lifelong learning.
- To create smart and ethical professionals and entrepreneurs who are recognized globally

# Revised Bloom's Taxonomy



# What will be the course outcome?

- Identify basic programming constructs and use the newly acquired skills to solve extensive programming problems.
- Discuss the mechanism of code reusability by creating own libraries of functions.
- Analyze the logic by designing code capable of passing various test cases.
- Interpret the principles of the object-oriented model and apply it in the implementation in C++ language.
- Categorize the theoretical knowledge and insights gained thus far to formulate working code
- Apply various programming constructs to build software applications



# Program Outcomes achieved from the course

---

- Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- Identify, formulate, review research literature, and analyses complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
- 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.

# Program Outcomes achieved from the course

---

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.

Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

# Program Educational Objectives achieved from the course

---

Apply acquired skills in software engineering, networking, security, databases, intelligent systems, cloud computing and operating systems to adapt and deploy innovative software solutions for diverse applications.

Apply diverse IT skills to design, develop, and evaluate innovative solutions for business environments, considering risks, and utilizing interdisciplinary knowledge for efficient real-time projects benefiting society.



## Unit I

**Concepts and Basics of C++ Programming** : Reading and writing data using cin and cout, Creating classes, Class objects, Accessing class members, Differences between Structures, Unions, Enumerations and Classes, Inline and Non-inline member functions, Static data members and static member functions, Differences between procedural and object oriented programming paradigms, Features of Input/output Streams

**Functions** : Functions with Default parameters/arguments, Inline Functions, Manipulator Functions, Function overloading and Scope rules, Friend of a class (friend function and friend class), Reference variables, Differences between Call by value, Call by address and call by reference, Recursion (Function, Member Function)

## Unit II

**Pointers, Reference Variables, Arrays and String Concepts** : Void pointer, Pointer arithmetic, Pointer to pointer, Possible problems with the use of pointers - Dangling pointer, Wild pointer, Null pointer assignment, Classes containing pointers, Pointer to objects, this pointer, Array of objects, The Standard C++ string class-defining and assigning string objects, Member functions, Modifiers of string class, Differences between pointer and reference variables, Array declaration and processing of multidimensional arrays(inside main and inside class), Pointer to data member

## Unit III

**Data File operations** : Opening and closing of files, Modes of file, File stream functions, Reading/Writing of files, Sequential access and random access file processing, Binary file operations, Classes and file operations, Structures and file operation

**Constructors, Destructors and File Handling** : Manager functions (constructors and destructor), Default constructor, Constructor with default arguments, Destructors, Parameterized constructor, Copy constructor , Initializer lists

## Unit IV

**Operator Overloading and Type Conversion** : Operator Overloading (unary operator, binary operator overloading), Type conversions - basic type to class type, class type to basic type

**Inheritance** : Inheritance Basics – derived class and base class, Types (simple, multi-level, multiple and hierarchical), Modes (private, protected, public inheritance), Overriding member functions, Order of execution of constructors and destructors, Resolving ambiguities in inheritance, Virtual base class, Aggregation and Composition.

## Unit V

**Dynamic Memory Management and Polymorphism** : Dynamic memory allocation using new and delete operators, Virtual destructors, Compile and run time polymorphism, Virtual functions, Dynamic constructors, Abstract classes and concrete class, Introduction to Self-Referential class, Pure virtual functions, Early binding and late binding, Memory leak and allocation failures

## Unit VI

**Exception Handling, Templates and Standard Template Library (STL)** : Basics of exception handling, Exception handling mechanism, Throwing mechanism, Catching mechanism, Rethrowing an exception, Function template and class template, Class template with inheritance, Introduction to STL- Containers, Algorithms and iterators, Container - Vector and List

# Course Assessment Model

<b>• CSE202</b>	<b>Marks break up*</b>
• Attendance	5
• Programming Practice	20
• CA (Two best out of Three CBTs)	30
• ETP (Practical /Laptop)	45
• <b>Total</b>	<b>100</b>

• **# No MTE**

# Academic Tasks

Academic Task	Tentative Week
CA-1: Programming Practice (MCQs + Coding) (Mandatory)	Week1 – Week14
CA-2: Mix of MCQs(10 Marks) + Coding Problems(20 Marks)	Week 5
CA-3: Mix of MCQs(10 Marks) + Coding Problems(20 Marks)	Week 9
CA-4: Mix of MCQs(10 Marks) + Coding Problems(20 Marks)	Week 12

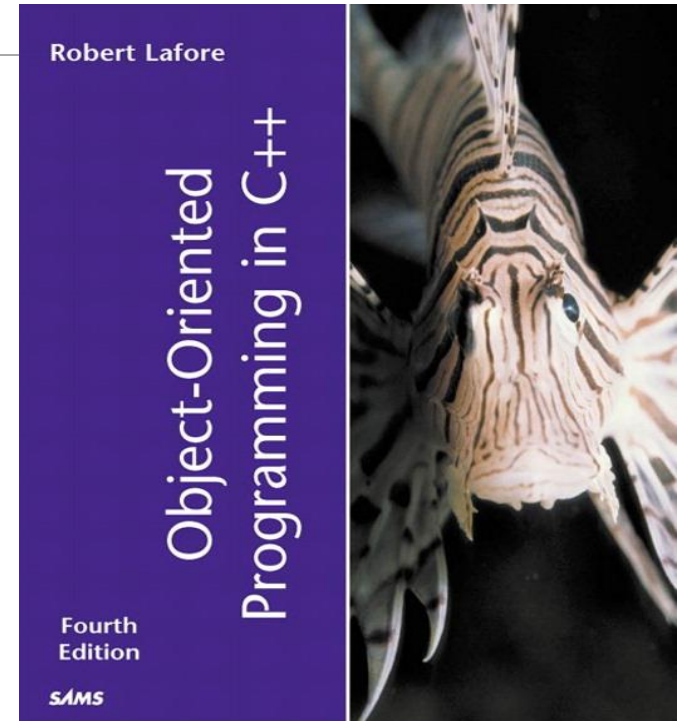
# Course Details

## Text Book

“OBJECT ORIENTED  
PROGRAMMING IN C++”

by

Robert Lafore





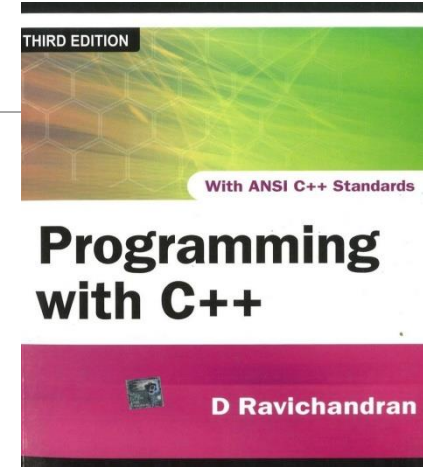
L P U

NAAC  
GRADE **A++**

# Reference Books

PROGRAMMING WITH C++

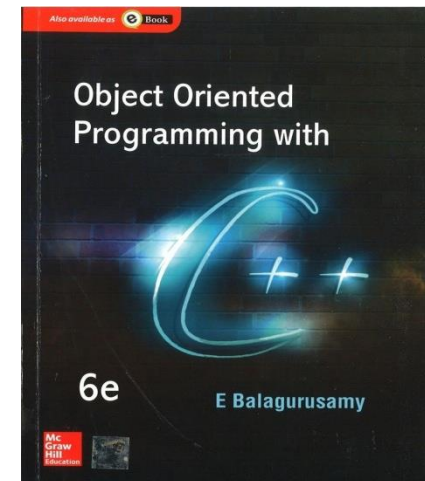
by D RAVICHANDRAN



OBJECT ORIENTED

PROGRAMMING IN C++

by E BALAGURUSAMY



# Practice Problems

---

- In the Practice Problems marks are only for coding Problems.
- MCQs are only for practice.

Unit wise Sequence locking

# Marks Calculation for Programming Practice

- In order to qualify for programming practice marks, the student must solve a minimum of 50% (50% Coding Problems + 50% MCQs) problems (eligibility condition)
- No marks for MCQs.

## Scoring Criteria:

The students will be given approx. 90 coding problems and 90 MCQs on the selected third party platform. The questions will be framed with equal distribution from the complete syllabus. In order to qualify for programming practice marks, the student should solve at least 50% of the coding problems and 50% of MCQ questions (eligibility condition).

# Marks Calculation for Programming Practice

---

The maximum marks out of 20 marks for which the student would be eligible for Programming Practice would be based on the Percentage of scored marks in the questions solved by the student.

The final marks for Programming Practice would be calculated by prorating the eligible marks for which the student is eligible (as explained in the above point) with the percentage of marks student has scored in the proctored Coding Contests conducted as CBT's (The final marks would be round up for the students).



## Marks Calculation for Programming Practice

---

Example – If a student solves 63(Correct) questions out of 90 questions (i.e. 70% questions solved) then the student would be eligible for 70% of 20 marks which is 14 marks. And the student has scored 24 out of 30 in the CBTs i.e. 80% marks in CBTs, his Programming practice final marks would be 80% of 14 marks that he was eligible for which is 11.2 rounded up to 12 marks out of 20 for Programming Practice.

# Daily Practice Problems

S. No	Unit	Question	Tentative Dates for Completion
1	Unit-1	At-least 15 Multiple Choice Questions and 15 Coding problems in each unit	1 <sup>st</sup> September 2024
2	Unit-2		22 <sup>nd</sup> September 2024
3	Unit-3		6 <sup>th</sup> October 2024
4	Unit-4		26 <sup>th</sup> October 2024
5	Unit-5		6 <sup>th</sup> November 2023
6	Unit-6		17 <sup>th</sup> November 2023

**Note:-** Most Important for the improvement of Performance in Course Assessments.

# End Term Practical (45 Marks)

---

- Online Assessment on third party Platform [50% ]  
[Mix of Coding Problem and MCQs]
- Viva [50%]

# The marks scored in the online assessment will be prorated according to the viva marks, if the student scores less than 60% in the viva.

# Why Star Course?

---

- Weightage in Gate/Govt. exams
- Industry demand
  - Product Based
  - Service Based
- Higher Studies
- Government Jobs

# Execution Strategies

Topic: Activities	Details of the Activities Planned	Is this Activity a part of Evaluation (Yes/No): Which CA/MTE/ETE	Tentative week of conduct of activity	Responsibility: Who will ensure the conduct of the activity (specify the plan for the same)	Expected Outcome	References	Quantification
First Continuous Assessment	The continuous assessment will be planned as a coding test consisting of 2 coding questions and 10 MCQs on an online judge.	Yes	5	The concerned subject teacher will ensure the smooth conduct of the activity.	Evaluation and reiteration of the taught concepts.	NA	
Second Continuous Assessment	The continuous assessment will be planned as a coding test consisting of 2 coding questions and 10 MCQs on an online judge.	Yes	10	The concerned subject teacher will ensure the smooth conduct of the activity.	Evaluation and reiteration of the taught concepts.	NA	

# Execution Strategies

## Details of Course Enrichment Activities

Third Continuos Assessment	The continuous assessment will be planned as a coding test consisting of 2 coding questions and 10 MCQs on an online judge.	Yes	12	The concerned subject teacher will ensure the smooth conduct of the activity.	Evaluation and reiteration of the taught concepts.	NA	
End Term Practical	Assessment will be conducted on the third-party platform consists of coding problems and MCQs, Followed by the viva by the neutral examiner	Yes	After 14 <sup>th</sup> week	Will be planned by the examination division	Evaluation and test the overall learning	NA	
Software Mapping	Dev C++/GCC/Online Compiler will be used for the live coding demonstration during the classes	Yes	1 <sup>st</sup> class onwards	The concerned subject teacher will use the platform from 1 <sup>st</sup> lecture onwards in their classes for reference	Hands on the software	NA	

# Execution Strategies

Use of third party platform for evaluation	All the assessments including ETP and Programming practice will be done on third party platform	Yes	Student will solve the coding problems and MCQs on the third party platform in all the assessments	The concerned subject teacher will ensure the smooth conduct of the activity.	Use of the third-party platform in assessment to improve the skills	NA
--	---	-----	--	---	---	----

## Details for Pedagogical Initiatives for the above-mentioned course

Lecture Number	Pedagogical Initiative	Expected Outcomes	References	Quantification
Lecture Number 1 onwards	Live demonstration of code and third-party platform use	The students will be well versed with coding concepts and will have the necessary knowledge how to write code.	Material available on the online platform	



## Evaluation strategy for all Components of the courses

Type of Assessment (Add rows in case some other assessments are applicable)	Type of Assessment e.g. Test/Term Paper/project etc.	Details of academic task	Parameters/Rubric of Evaluation	External/Internal	Details of External Evaluations	References	Quantification
Coding Contest	Test	The continuous assessment will be planned as a coding test consisting of 2 coding questions and 10 MCQs on an online platform.	The code will be auto evaluated by an online platform	External online platform	The code written by the student will be evaluated on the online platform.	NA	
Programming Practice	Test	The student will be asked to attempt approx. 90 problems with 15 problems per unit. Problems will open unit wise.	The problems will be auto evaluated by an online platform	External online platform	The code written by the student will be evaluated on the online platform.	NA	



# Cohort

---

## Software Development

- Product based
- Service based



LPU

NAAC  
GRADE **A++**

# MOOCs

---

---

Not Applicable

# Skill Set

---

Programming Skills

Logic building

Code analysis



# Platform Used

---

Live Demonstration of selected **Third Party** Platform

# The hitch.....

---

Some burning questions in mind.....

Is C the only language behind the development of all these powerful softwares

Why C++?



How C++ is different from C language?

Is there no scope of this language in industry?

# Flashback.....

What C language can do?????

- Device drivers are 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
- Embedded systems are also developed with the help of C language

# Is C only language?????

Lets answers your questions

---

- No, C is not the only language for the development of all these powerful system
- If we talk about Windows operating system, only Kernel is written in C language rest all other features are provided with the help of C++ language



# How does it differ from C

---

- C uses top down approach for problem solving
- There are no means of providing security.
- C doesn't support inheritance, makes it more complex to use because everything has to be written from scratch.
- C is not able to represent real world modelling



# No scope of language in Industry

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



Microsoft



amazon



NVIDIA®

Google



vmware®



redhat



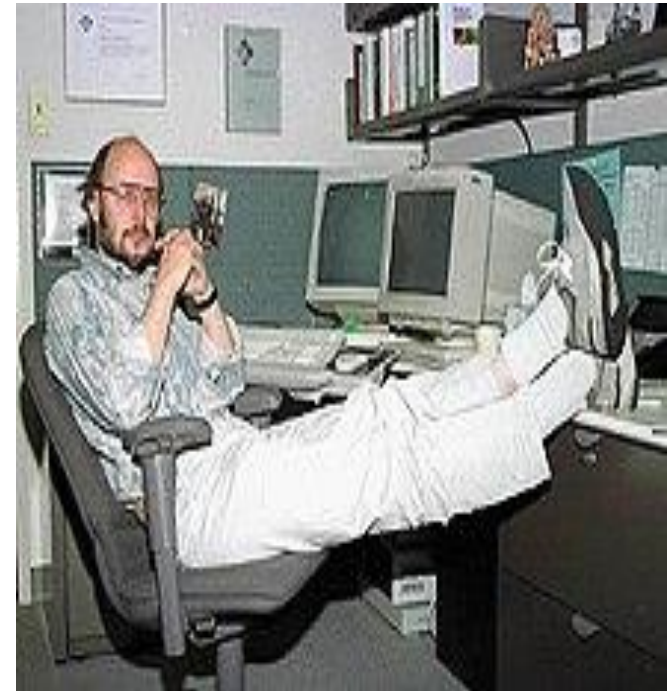
memsql

YAHOO!



# Introduction to C++

- In 1979, Bjarne Stroustrup, began work on "C with Classes"
- The motivation for creating a new language originated from Stroustrup's experience in programming for his Ph.D. thesis.
- Stroustrup found that Simula had features that were very helpful for large software development, but the language was too slow for practical use, while BCPL was fast but too low-level to be suitable for large software development.



•Git



•Microsoft Excel



•Oracle Database



•MySql



•Linux



•Unix



•Android



•Google



---

# WHY C++?????

Lets see



# Key features of C++

## ➤ Object and Classes

---

Real world modelling

## ➤ Encapsulation

Wrapping

## ➤ Data Abstraction

Hiding of non required information

## ➤ Inheritance

Reusability

# Key features of C++

---

➤ Polymorphism

Many forms

➤ Data Hiding

Security

➤ Message Passing

Communication

# (Objects) Daily Analysis

Look around you and identify some objects



**Everything is an OBJECT**



# Object ,Object and Object

A student, a professor

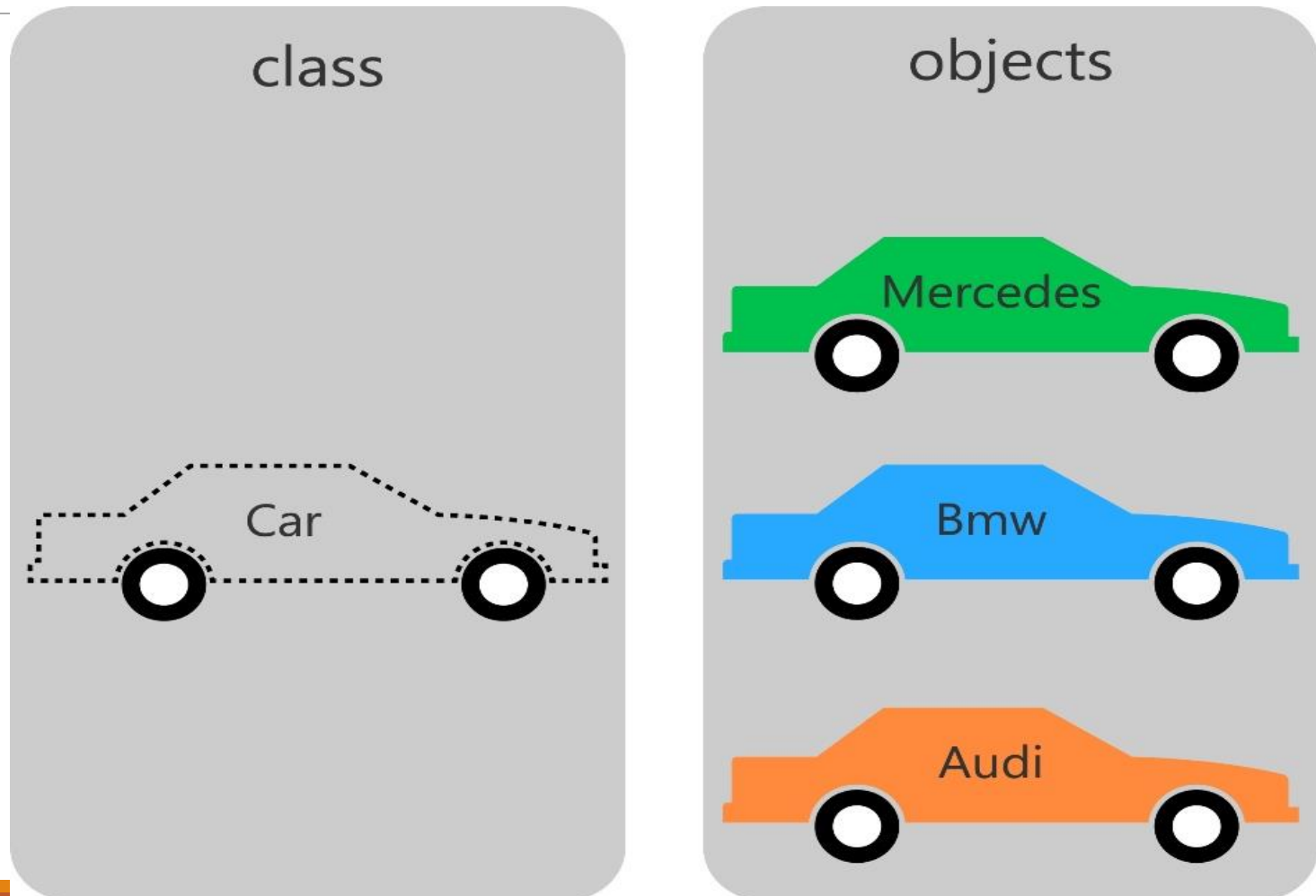
A desk, a chair, a classroom, a building

A university, a city, a country

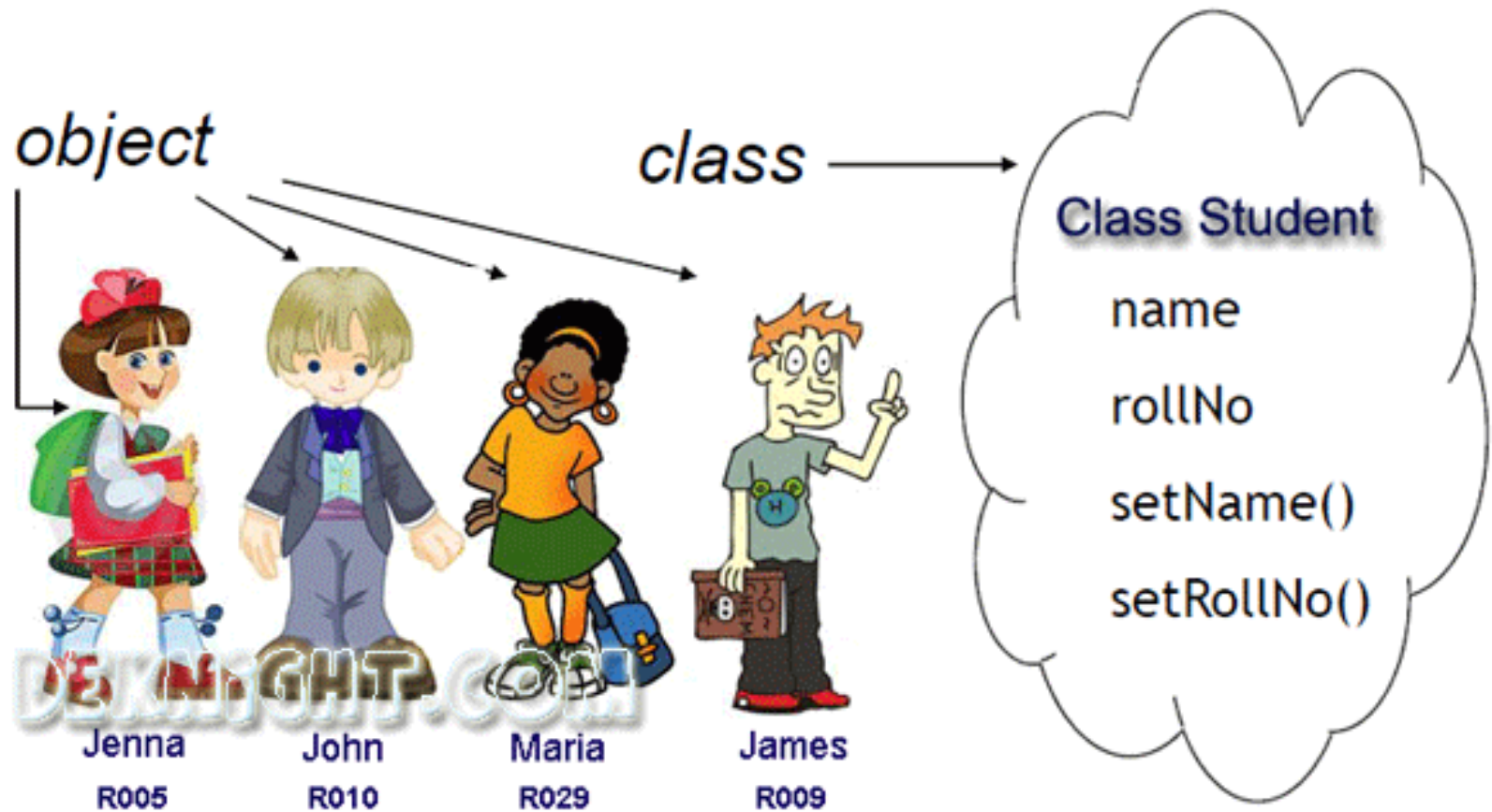
A subject such as CS, Math, History, ...



# Classes and Objects(Example 1)



# Classes and Objects(Example 2)



# Classes and Objects

---

A *class* is a prototype or blueprint for creating objects

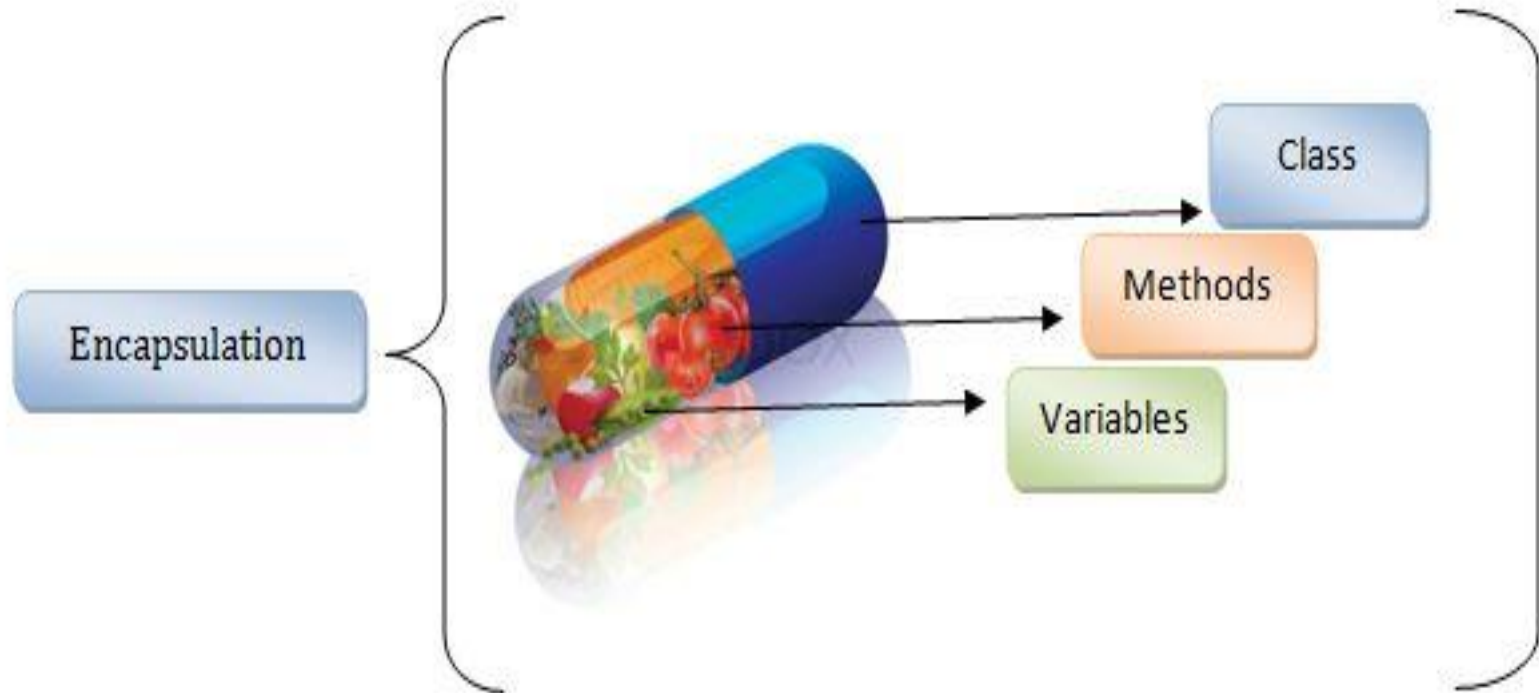
When we write a program in an object-oriented languages like C++,Java, we define classes, which in turn are used to create objects

# Features Continued...

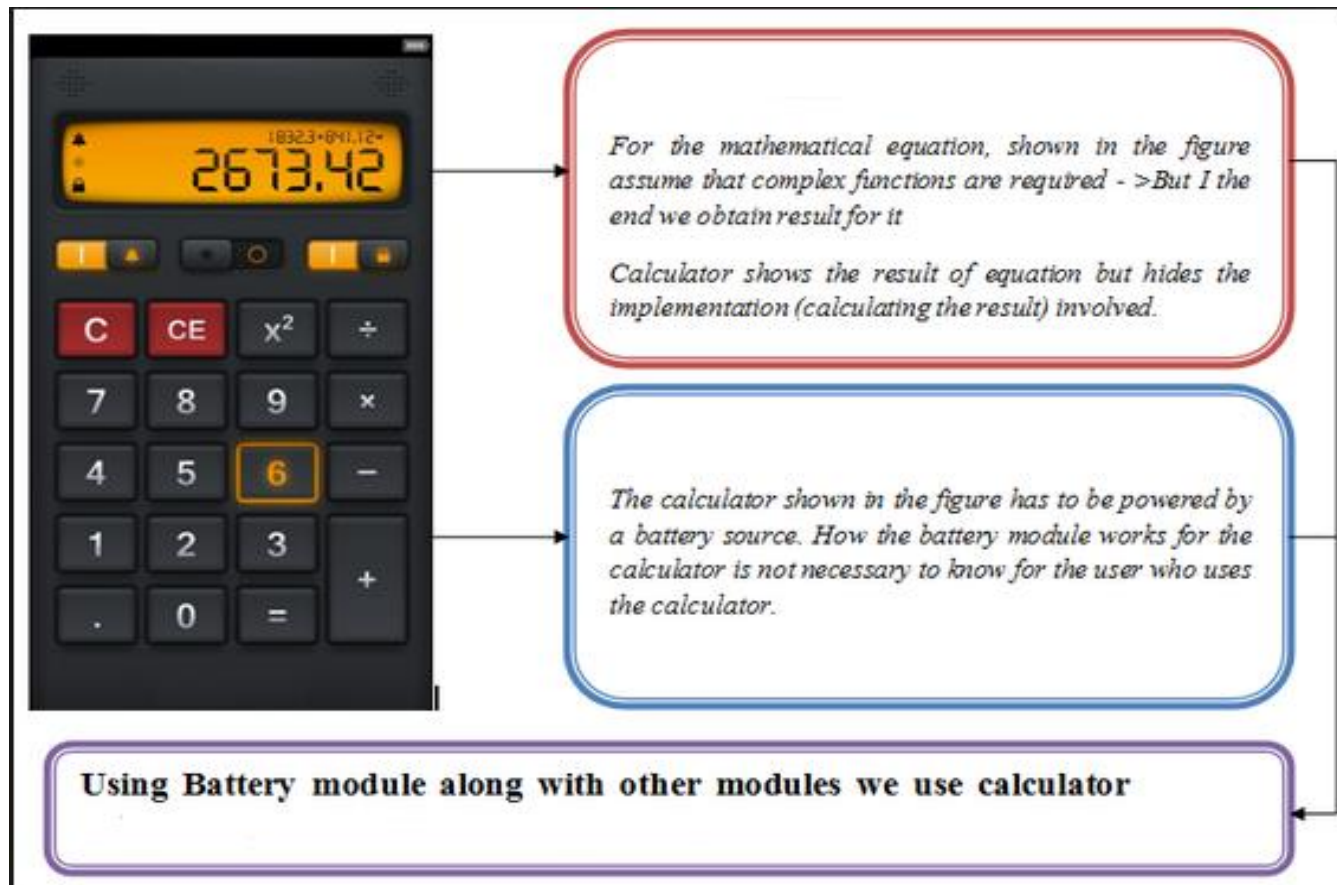
---

- ☐ Encapsulation
- ☐ Abstraction
- ☐ Reusability
- ☐ Inheritance
- ☐ Polymorphism

# Encapsulation



# Abstraction(Hiding Details)



# Reusability

Existing Features

Existing Features + Additional Features



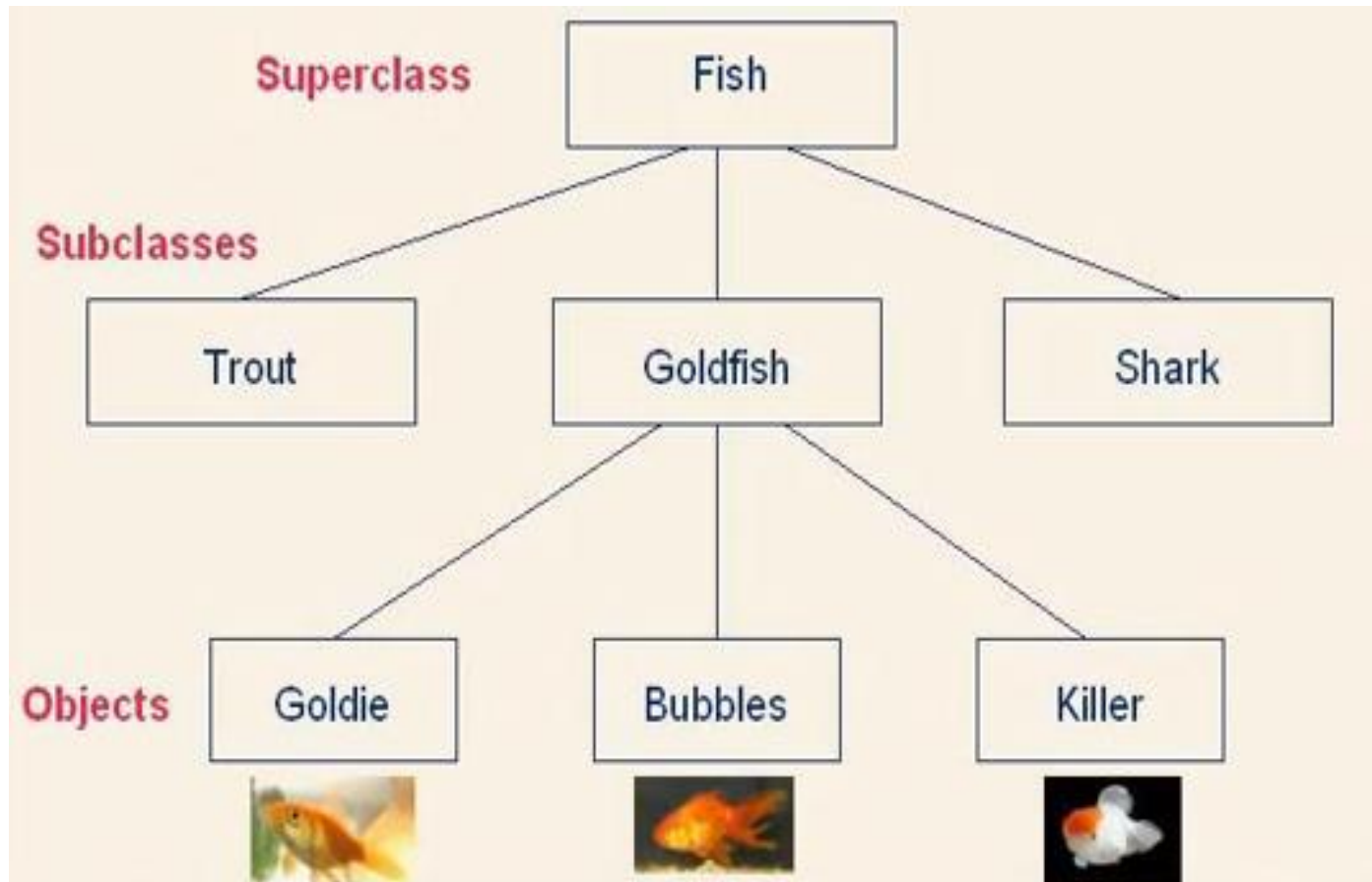
# Inheritance



But Mummy,  
where did my  
blue eyes  
come from?



# Inheritance....



# Polymorphism

**In class**



**In CCD**

**In front of father**



**One thing and many forms**

# Polymorphism

Ability to appear in many forms...

If you ask different animal to “speak”, they responds in their own way.



Same Function Different Behavior

# Course Contents

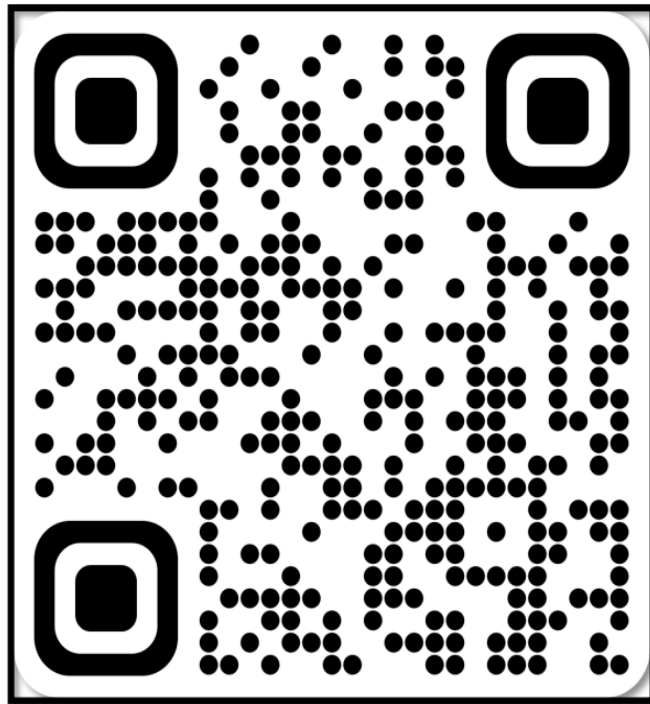
---

- Classes and Objects
- Functions
- Pointer
- Array
- String
- Constructor and Destructor
- File Handling
- Operator Overloading
- Type Conversion
- Inheritance
- DMA
- Polymorphism
- Exception Handling
- Templates and STL

## Zero Lecture - Feedback

Please refrain from including your name or registration number in your feedback. Your honest opinions shall be valued and appreciated.

**Scan Me**



or visit URL:

<https://forms.gle/KViuXpJEtN2rzG5A9>



---

# Thank you!

---