

**Program Name** : Computer Engineering Program Group  
**Program Code** : CO/CM/IF/CW  
**Semester** : Third  
**Course Title** : Object Oriented Programming using C++  
**Course Code** : 22316

### 1. RATIONALE

In the modern world of Information technology, the Object Oriented Programming has become the most preferred approach for software development. It offers a powerful way to cope up with complexity of real world problems. Among the OOP languages available, C++ is the primitive language which develops fundamental understanding of Object Oriented Concepts. This course enables students to develop programs in 'C++' using Object Oriented Programming approach.

### 2. COMPETENCY

The aim of this course is to help the student to attain the following industry identified competency through various teaching learning experiences:

- **Develop applications Using OOPs concepts in C++.**

### 3. COURSE OUTCOMES (COs)

The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following *industry oriented* COs associated with the above mentioned competency:

- Develop C++ programs to solve problems using Procedure Oriented Approach.
- Develop C++ programs using classes and objects.
- Implement Inheritance in C++ program.
- Use Polymorphism in C++ program.
- Develop C++ programs to perform file operations.

### 4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme			Credit (L+T+P)	Examination Scheme												
L	T	P		Theory						Practical						
				Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
Max	Min	Max	Min		Max	Min	Max	Min	Max	Min	Max	Min	Max	Min		
3	2	2	7	3	70	28	30*	00	100	40	25@	10	25	10	50	20

(\*): Under the theory PA; Out of 30 marks, 10 marks of theory PA are for micro-project assessment to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessment of the UOs required for the attainment of the COs.

**Legends:** L-Lecture; T – Tutorial/Teacher Guided Theory Practice; P - Practical; C – Credit, ESE - End Semester Examination; PA - Progressive Assessment

### 5. COURSE MAP (with sample COs, PrOs, UOs, ADOs and topics)

This course map illustrates an overview of the flow and linkages of the topics at various levels of outcomes (details in subsequent sections) to be attained by the student by the end of the

course, in all domains of learning in terms of the industry/employer identified competency depicted at the centre of this map.

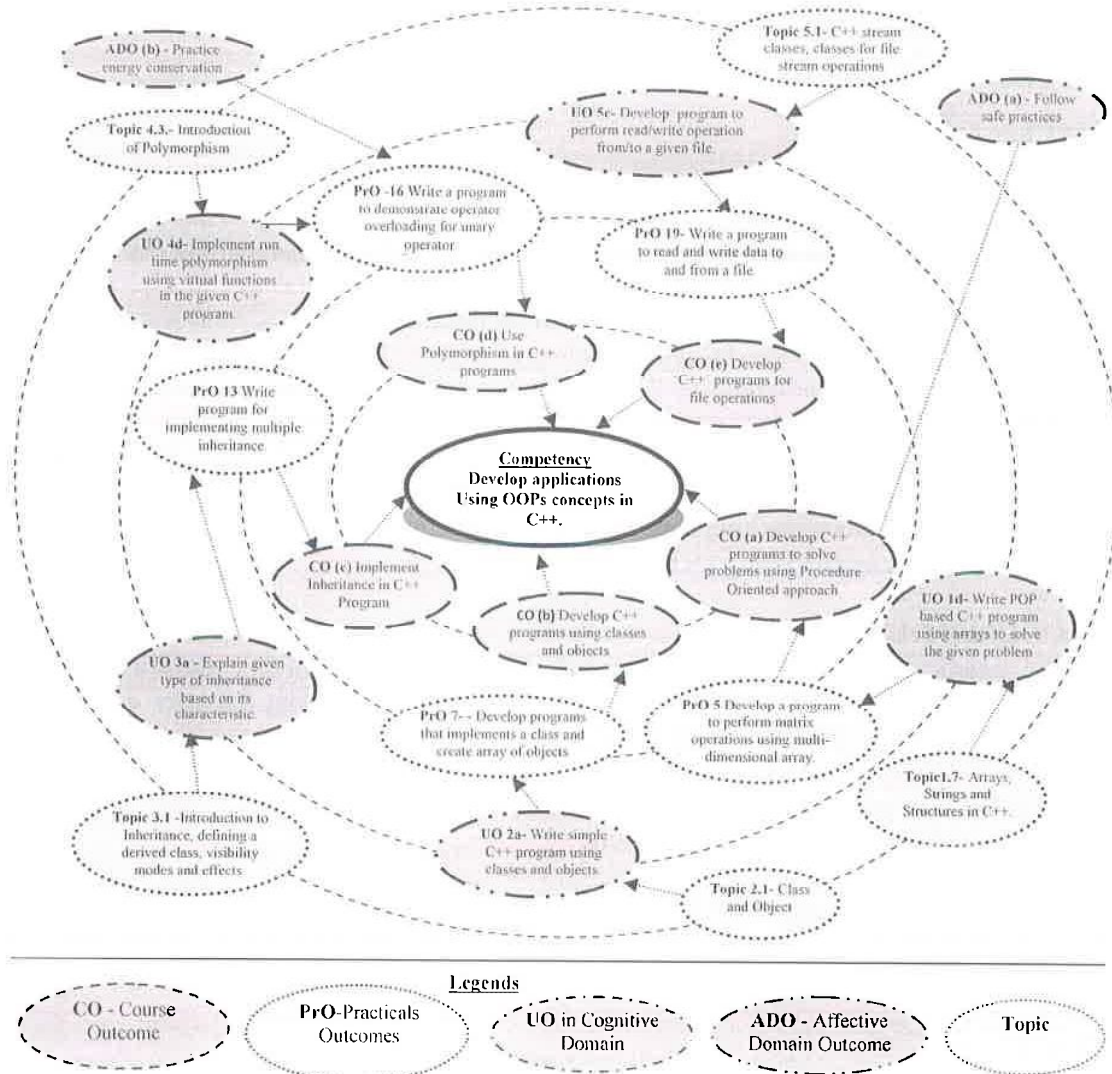


Figure 1 - Course Map

**6. SUGGESTED PRACTICALS/ EXERCISES**

The practicals in this section are PrOs (i.e. sub-components of the COs) to be developed and assessed in the student for the attainment of the competency:

Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1	Develop minimum 2 programs using constants, variables, arithmetic expression, operators, exhibiting data type conversion.	1	02*
2	Develop a program to implement decision making statements (If-else, switch).	1	02
3	Develop a program to demonstrate control structures (for, while, do-while).	1	02



Sr. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
4	Develop a program to implement 1-dimension array.	I	02 *
5	Develop a program to perform matrix operations using multi-dimensional array.	I	02
6	Develop programs that implements a class and use it with objects.	II	02*
7	Develop programs that implements a class and create array of objects.	II	02*
8	Write a program to implement friend function.	II	02*
9	Write a program to implement inline function.	II	02
10	Write a program to implement all types of constructors (constructor overloading) with destructor.	II	02*
11	Write a program for implementing single inheritance	III	02*
12	Write a program for implementing multi level inheritance.	III	02
13	Write a program for implementing multiple inheritance.	III	02*
14	Develop minimum 1 program to demonstrate Pointer to object.	IV	01 *
15	Develop minimum 1 program to demonstrate Pointer to derived class	IV	01 *
16	Write a program to demonstrate operator overloading for Unary operator.	IV	02
17	Write a program to demonstrate operator overloading for Binary operator	IV	02
18	Write a program to demonstrate function overloading	IV	02*
19	Write a program to read and write data to and from a file.	V	02
<b>Total</b>			<b>38</b>

**Note**

- i. A suggestive list of **PrOs** is given in the above table. More such PrOs can be added to attain the COs and competency. A judicious mix of minimum 12 or more practical need to be performed, out of which, the practicals marked as "\*" are compulsory, so that the student reaches the 'Precision Level' of Dave's 'Psychomotor Domain Taxonomy' as generally required by the industry.
- ii. The 'Process' and 'Product' related skills associated with each PrO is to be assessed according to a suggested sample given below:

S. No.	Performance Indicators	Weightage in %
a.	Correctness of algorithm	40
b.	Debugging ability	20
c.	Quality of input and output displayed (messaging and formatting)	10
d.	Answer to sample questions	20
e.	Submit report in time	10
<b>Total</b>		<b>100</b>

The above PrOs also comprise of the following social skills/attitudes which are Affective Domain Outcomes (ADOs) that are best developed through the laboratory/field based experiences:



- a. Follow safety practices.
- b. Practice good housekeeping.
- c. Demonstrate working as a leader/a team member.
- d. Follow ethical practices.

The ADOs are not specific to any one PrO, but are embedded in many PrOs. Hence, the acquisition of the ADOs takes place gradually in the student when s/he undertakes a series of practical experiences over a period of time. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- 'Valuing Level' in 1<sup>st</sup> year
- 'Organising Level' in 2<sup>nd</sup> year and
- 'Characterising Level' in 3<sup>rd</sup> year.

#### 7. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

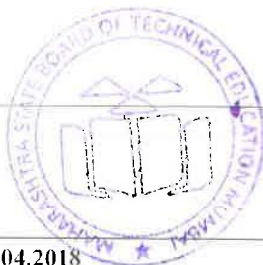
The major equipment with broad specification mentioned here will usher in uniformity in conduct of experiments, as well as aid to procure equipment by authorities concerned.

S. No.	Equipment Name with Broad Specifications	PrO S. No.
1	Computer system (Any computer system with basic configuration)	All
2	'C++' Compiler (Turbo C++ compiler/GCC compiler or any other C++ compiler)	

#### 8. UNDERPINNING THEORY COMPONENTS

The following topics are to be taught and assessed in order to develop the sample UOs given below for achieving the COs to attain the identified competency. More UOs could be added:

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
<b>Unit – I Principles of Object Oriented Programm ing</b>	1a. Write simple C++ program for solving the given expression using POP approach. 1b. Write POP based C++ program using decision making and loop structure for the given situation. 1c. Write POP based C++ program using arrays to solve the given problem. 1d. Use the structure in C++ program for solving the given problem.	1.1 Procedure Oriented Programming (POP) verses Object Oriented Programming (OOP), 1.2 Basic concepts of Object Oriented Programming, Object Oriented Languages, Applications of OOP. 1.3 C verses C++, Structure of C++ program, Simple C++ Program. 1.4 Tokens, keywords, variables, constants, basic data types, User defined data types, type casting, operators, expressions. 1.5 Control structures: Decision making statements and Loops 1.6 Scope resolution operator, memory management operators. 1.7 Arrays, Strings and Structures in C++.





Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
<b>Unit– II Classes and Objects</b>	2a. Develop relevant friend functions to solve the given problem. 2b. Write C++ program to use array of given objects. 2c. Write C++ program to create the given object using constructor. 2d. Write program to delete the given object using destructor in C++ program.	2.1 Class & Object: Introduction, specifying a class, access specifiers, defining member functions, creating Objects, memory allocations for objects. 2.2 Static data members, static member function, friend Function 2.3 Array of Objects, Object as function arguments. 2.4 Concepts of Constructors, Types of constructors. 2.5 Multiple Constructors in a Class, Constructors with default arguments. 2.6 Destructors.
<b>Unit-III Extending classes using Inheritance</b>	3a. Explain given type of inheritance based on its characteristic. 3b. Implement given type of inheritance in C++ program. 3c. Write C++ program using virtual base class. 3d. Use constructor in the given derived class.	3.1 Introduction to Inheritance, defining a derived class, visibility modes and effects. 3.2 Types of Inheritance : Single, multilevel, multiple, hierarchical, hybrid 3.3 Virtual base class, abstract class, constructors in derived class.
<b>Unit –IV Pointers and Polymorphism in C++</b>	4a. Create C++ programs to perform the given arithmetic operations using pointers. 4b. Use function overloading to solve the given problem 4c. Use operator overloading to solve the given problem 4d. Implement run time polymorphism using virtual functions in the given C++ program.	4.1 Concepts of Pointer: Pointer declaration, Pointer operator, address operator, Pointer arithmetic. 4.2 Pointer to Object: Pointer to Object, this pointer, Pointer to derived class. 4.3 Introduction of Polymorphism, Types of Polymorphism. 4.4 Compile time Polymorphism: Function overloading, operator overloading, overloading of unary and binary operators, Rules for operator overloading. 4.5 Run time polymorphism: Virtual functions, rules for virtual functions, pure virtual function
<b>Unit-V File operations</b>	5a. Identify relevant class for performing the given file operation. 5b. Write statement to open and close the given file in C++. 5c. Develop C++ program to perform read/write operation from/to the given file.	5.1 C++ stream classes, Classes for file stream operations. 5.2 Opening files, closing files, reading from and writing to files. 5.3 Detection of end of file, file modes.



*Note: To attain the COs and competency, above listed UOs need to be undertaken to achieve the 'Application Level' of Bloom's 'Cognitive Domain Taxonomy'.*

## 9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Principles of Object Oriented Programming	08	2	4	8	14
II	Classes and Objects	14	2	4	12	18
IV	Inheritance: Extending classes	10	2	4	10	16
V	Pointers and Polymorphism in C++	10	-	4	10	14
VI	Working with files	06	-	2	6	08
<b>Total</b>		<b>48</b>	<b>6</b>	<b>18</b>	<b>46</b>	<b>70</b>

**Legends:** R=Remember, U=Understand, A=Apply and above (Bloom's Revised taxonomy)

**Note:** This specification table provides general guidelines to assist student for their learning and to teachers to teach and assess students with respect to attainment of UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table.

## 10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related *co-curricular* activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct following activities in group and prepare reports of about 5 pages for each activity, also collect/record physical evidences for their (student's) portfolio which will be useful for their placement interviews:

- Prepare journal of practicals.
- Undertake micro-projects using Object Oriented Concepts.

## 11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various learning outcomes in this course:

- Massive open online courses (*MOOCs*) may be used to teach various topics/sub topics.
- 'L' in item No. 4 does not mean only the traditional lecture method, but different types of teaching methods and media that are to be employed to develop the outcomes.
- About **15-20% of the topics/sub-topics** which is relatively simpler or descriptive in nature is to be given to the students for *self-directed learning* and assess the development of the COs through classroom presentations (see implementation guideline for details).
- With respect to item No.10, teachers need to ensure to create opportunities and provisions for *co-curricular activities*.
- Guide student(s) in undertaking micro-projects.
- Demonstrate students thoroughly before they start doing the practice.
- Encourage students to refer different websites to have deeper understanding of the subject.
- Observe continuously and monitor the performance of students in Lab.

## 12. SUGGESTED MICRO-PROJECTS

**Only one micro-project** is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based. However, in the fifth and sixth semesters, it should be preferably be **individually** undertaken to build up the skill and confidence in every student to become problem solver so that s/he contributes to the projects of the industry. In special situations where groups have to be formed for micro-projects, the number of students in the group should **not exceed three**.

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The total duration of the micro-project should not be less than **16 (sixteen) student engagement hours** during the course. The student ought to submit micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects are given here. Similar micro-projects could be added by the concerned faculty:

- Develop library management application.
- Develop hotel management application.
- Develop bank management application.
- Develop store management application.
- Develop hospital management application.
- Any other micro-projects suggested by subject faculty on similar line.  
(Use Object Oriented concepts and may also use file handling features of 'C++' to develop above listed applications)

### 13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication
1	Object Oriented Programming with C++	Balgurusamy, E.	McGraw Hill Education, New Delhi 2015, ISBN: 9781259029936
2	The C++ Programming Language	Stroustrup, B.	Pearson Education, New Delhi 2015, ISBN: 9780201889543
3	Object Oriented Programming in C++	Lafore, R.	Sams Publication, New Delhi 2015, ISBN: 9780672323089
4	C++ The Complete Reference	Schildt, H.	McGraw Hill Professional, New Delhi 2015, ISBN: 9780072226805
5	Object Oriented Programming in C++	Subburaj, R.	Vikas Publication, New Delhi 2015, ISBN: 9789325969964
6	C++ Programming	Dr. Rajendra Kawale	Devraj Publications

### 14. SUGGESTED SOFTWARE/LEARNING WEBSITES

- [https://www.tutorialspoint.com/cplusplus/cpp\\_object\\_oriented.htm](https://www.tutorialspoint.com/cplusplus/cpp_object_oriented.htm)
- <http://www.studytonight.com/cpp/cpp-and-oops-concepts.php>
- [https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp3\\_OOP.html](https://www3.ntu.edu.sg/home/ehchua/programming/cpp/cp3_OOP.html)
- <https://www.hscripts.com/tutorials/cpp/cpp-oops-concepts.php>
- <https://www.khanacademy.org/>
- <http://www.nptel.ac.in>

