



**B. M. S. INSTITUTE OF TECHNOLOGY AND
MANAGEMENT
YELAHANKA, BANGALORE-064
Department of Computer Science & Engineering**

OPEN COURSES ON
“OOP Concepts using C++”
&
“OOP Concept using C++ and QT Framework”

An Open course on “OOP Concept using C++” and “OOP Concept using C++ and QT Framework” by the department of CSE during 12th to 16th Feb 2019. It was open for all branches and all semester students. A total of 94 students registered for the courses. The resource persons for the open courses were internal staff Mr. Vishwa Kiran and Dr. Satish from the department of CSE. The course was focused and planned exclusively for fourth semester students since VTU says to conduct a bridge course to fill the gap in syllabus. Also, to help the students gain the fundamental knowledge about OOP concepts. The resource person tried to bring out “To know what students do not know”. Also, it would help the placement exams of the students of various branches.

Resource Persons:

Mr. Vishwa Kiran, Asst Prof, Dept of CSE (For all 5 days)

Mr. Satish Kumar, Asst Prof, Dept of CSE (One session in parallel)

Faculty Coordinators	Responsibilities
Bharathi R, Associate Professor, Dept. of CSE	Designing the online survey form, taking feedback.
Mari Kirthima A, Assistant Professor, Dept. of CSE	Taking attendance
Ravi Kumar B N, Assistant Professor, Dept. of CSE	Hospitality, Memento design and procurement
Durga Bhavani A, Assistant Professor, Dept. of CSE	Registration
Jagadish P , Assistant Professor, Dept. of CSE	Certificate design and printing
HemaMalini B H, Associate Professor, Dept. of CSE	Overall coordination – Purchase of registration kits, budgeting, approvals, account settlements, report generation, thanks giving.

Registration Details:

Department	Registration Count
CSE	88
ISE	02
ECE	02
TCE	02
Total	94

Topics Covered:

Software Setup:

- ❖ **Complete Installation of Linux with g++ compiler**
Or
- ❖ **Microsoft visual studio 2010**

Course contents

Day 1

- Object Oriented Methodology
 - ❖ **The transition from procedural to an object-oriented approach**
 - ❖ **The procedural approach**
 - ❖ **Functional decomposition**
 - ❖ **Interface and implementation**
 - ❖ **The Object-oriented approach**

- Object-Oriented Concepts
 - ❖ **Abstraction**
 - ❖ **Encapsulation**
 - ❖ **Modularity**
 - ❖ **Interface and implementation**
 - ❖ **Classes & Objects – the different perspectives**
 - ❖ **State, Identity, and Behavior**
 - ❖ **Inheritance & Class Hierarchy**
 - ❖ **Polymorphism: Early v/s late binding**

- **The C++ language basics**
 - ❖ **A simple C++ program**
 - ❖ **Illustration of headers and implementation**
 - ❖ **The three file approach**

Day 2

- **Functions**
 - ❖ **Inline function**
 - ❖ **Function prototype**
 - ❖ **Recursive Function**
 - ❖ **Overloading Functions**
 - ❖ **Pass by value v/s Pass by ref**
 - ❖ **Default function Arguments**
 - ❖ **Variable Parameter List**
 - ❖ **Temporary objects**

- **Class, Object and references concepts**
 - ❖ **Class declaration & object mechanism**
 - ❖ **Access specifiers**
 - ❖ **Separating interface from implementation**
 - ❖ **Memory allocation for objects**
 - ❖ **Managing the heap**
 - ❖ **Array of objects**
 - ❖ **Accepting/Returning object references**

Day 3

- **Constructors and Destructors**
 - ❖ **Objects & references**
 - ❖ **Constructors and Destructors**
 - ❖ **Characteristics of constructors**
 - ❖ **Overloaded constructors**
 - ❖ **Copy constructor**
 - ❖ **Destructors**
 - ❖ **Use of constructors**
 - ❖ **Passing objects**
 - ❖ **Copy constructor**
 - ❖ **Overloading copy constructor**

- **Data Members**
 - ❖ **Static members and objects on the heap**
 - ❖ **Static members of a class**
 - ❖ **Static data members**
 - ❖ **Static member functions**
 - ❖ **A discussion on static members**
 - ❖ **Objects on the heap**
 - ❖ **The this pointer**
 - ❖ **Constant members**
 - ❖ **Constant data members**
 - ❖ **Constant member functions**
 - ❖ **Constant objects**

- **Friend Functions**
 - ❖ **Friend and classes**
 - ❖ **Object communication**
 - ❖ **Friend functions**
 - ❖ **Friend classes**

Day 4

- **Operator Overloading**
 - ❖ **The operator function**
 - ❖ **Operators that cannot be overloaded**
 - ❖ **Overloading unary operators**
 - ❖ **Overloading Binary operators**
 - ❖ **Using member functions & friend functions –**
 - ❖ **when to use what**
 - ❖ **Interpreting the operator function call**
 - ❖ **Function overloading the operator function**
 - ❖ **The assignment operator**
 - ❖ **Returning objects & assignment operator**
 - ❖ **Overloading >>&<< operators**

- **Code Reusability Techniques using Inheritance & containment**
 - ❖ **Containment vs Inheritance**
 - ❖ **Private vs protected inheritance**
 - ❖ **Hybrid Inheritance / virtual base class**
 - ❖ **Changing scope of base member in derived class**
 - ❖ **The inheritance mechanism**
 - ❖ **Types of inheritance**
 - ❖ **Single level, multi-level, multiple, hierarchical, hybrid**
 - ❖ **Inheritance – is it allocation or accessibility**
 - ❖ **The protected access specifier**
 - ❖ **Inheritance in private, protected & public mode**
 - ❖ **Constructors & destructors in a derived class**
 - ❖ **Multiple inheritance**
 - ❖ **Why a constructor is not inherited**
 - ❖ **Is-a & Has-a relationship**

- ❖ Nested classes & Containership
- ❖ Pointers & Classes
- ❖ Pointers to Class Members

Day 5

- Virtual functions
 - ❖ Early binding vs late binding (Dynamic Polymorphism)
 - ❖ Need for Virtual Function
 - ❖ Virtual member function
 - ❖ Virtual pointer and virtual table



Handing over goodies



Hands on Session



Giving a Memento to the Resource Person



A session in seminar hall



A session in seminar hall

Coordinator
HemaMalini B H

H.O.D
Dr. Anil G N