

## **EMBEDDED SYSTEMS & ROBOTICS :LEVEL 2**

**Note:** <sup>(P)</sup> stands for practical classes along with theory.

### **MODULE I: BASICS OF “C++”**

#### **CHAPTER – 1 INTRODUCTION TO C++**

- Objective
- The history of c++
- Compilation steps
- Advantages and pretensions of c++
- Various programming technique
- Difference between structure programming language & object oriented programming language.
- Definition of oop
- Features of oops are the following :
  - Data abstraction
  - Encapsulation
  - Polymorphism
  - Inheritance
  - Exception handling
- A simple c++ program
- Stream
  - Standard input stream
  - Standard output stream
- Generic classes
- Templates
- How c++ compilation works
- Commenting your programs
  - Types of comments
  - Using comments
- Variables
  - Simple input / output constructs
  - Variable types
  - Names
  - New operator
  - Delete operator
- Operators
  - Arithmetic operators
  - Relational operators

- Logical operators
- Bitwise operators
- Increment / decrement operators
- Assignment operator
- Difference between c & c++
- Difference between delete & free
- Difference between new & malloc
- Summary
- Exercise
- Long question
- References

## CHAPTER 2 : CLASSES AND OBJECTS

- Objective
- Difference between structure & class
- Abstract data type
- Generic abstract data types
- Methods
- Classes
- Features of a class
- Syntax of a class
- Access specifiers
- Example of class
- Objects
- Examples of objects
- Creation of objects
- Accessing class members
- Program
- Abstraction
- How types of abstraction differs
- Reasons for the need of abstraction
- Encapsulation
- Features of encapsulation
- State
- Behavior
- Identity
- Constructors
- Types of constructor

- Destructors
- General syntax of destructors
- Instantiation of object
- Garbage collection
- What does libgc do?
- Common memory problems
- Memory leaks
- Premature frees
- Dynamic memory allocation
- What are memory management operators?
- New operator
- Delete operator
- Abstract classes
- Summary
- Exercise
- Long question
- References

### CHAPTER 3 : INHERITANCE

- Objective
- Definition (inheritance )
- Types of inheritance are as follows
- Single inheritance
- Multiple inheritance
- Hierarchical inheritance – one super class, many sub classes
- Multi level inheritance – derived from a derived class
- Hybrid inheritance – more than two types of inheritance
- Class hierarchy
- Derivation – public , private & protected
- Visibility mode
- Aggregation
- Composition
- Generalization
- Summary
- Exercise
- Long question
- References

#### CHAPTER 4 : POLYMORPHISM

- Objective
- Definition
- Features and advantages of the concept of polymorphism
- Types of polymorphism
- Virtual function
- Example of virtual function
- Rules for virtual function
- Pure virtual function
- Parametric polymorphism
- Overloadable operators
- Operators that cannot be overloaded
- Summary
- Exercise
- Long question
- References

#### CHAPTER 5 : GENERIC FUNCTION

- Objective
- Generic functions
- Class template with multiple parameters
- Example of template class with two generic data types
- Summary
- Exercise
- Long question
- References

#### CHAPTER 6 : STREAMS AND FILES

- Objective
- C++ streams
- Standard input (stdin)
- Standard output (stdout)
- Standard error (stderr)
- C++'s predefined streams
- Unformatted i/o operators
- Formatted console i/o operations

- Ios class functions and flags
- Defining display field width
- Setting precision
- Filling and padding
- Ios formatting flags
- Manipulators
- Custom / user – defined manipulators
- Space.cpp
- What is a file?
- Hierarchy of files stream classes
- Opening and closing a file
- Opening and closing of files explicitly
- Files modes
- Ascii and binary files
- Binary file
- Write ( ) and read ( ) functions
- Summary
- Exercise
- Long question
- References

#### **CHAPTER 7 : NAMESPACE , EXCEPTION HANDLING & STL**

- Objective
- Container adapters
- Algorithms
- Iterators
- Associative containers
- Sets and multisets
- Maps and multimaps
- Namespace definition

- Namespace fundamentals
- Using
- Unnamed namespace
- Exception handling fundamentals
- Exception handling constructs
- Using multiple catch statements
- Rethrowing an exception
- Introduction to the stl (standard template language)
- Containers
- Sequence containers
- Associative containers
- Member functions
- Container adapters
- Algorithms
- Iterators
- Sequence containers
- Vectors
- Lists
- Deques
- Associative containers
- Sets and multisets
- Maps and multimaps
- Summary
- Exercise
- Long question
- Reference

## **MODULE II: AVR**

### **AVR: (8hrs)**

- Introduction
- Feature
- Architecture
- Input/output port
- Timer
- Serial communication
- Interrupt
- ADC

### **HARDWARE INTERFACING <sup>(P)</sup>: (8hrs)**

- LED
- Seven segment
- Switches
- Sensors
- Motors
- Relay
- Buzzer
- LCD
- Keypad
- ADC
- RF module

### **MODULE III: PIC: (8hrs)**

- Introduction
- Feature
- Architecture
- Input/output port
- Timer
- Serial communication
- Interrupt
- ADC

### **HARDWARE INTERFACING <sup>(P)</sup>: (8hrs)**

- LED
- Seven segment
- Switches
- Sensors
- Motors
- Relay
- Buzzer
- LCD
- Keypad
- ADC
- RF module

Student's signature:

### **PROJECT: (5 days)**

ANY ONE OF THE CONTROLLER (AVR, PIC) IT DEPENDS UPON STUDENT