



MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY & MANAGEMENT

(Approved by AICTE-New Delhi, Accredited by **NAAC** with 'A' & Affiliated to JNTU, Hyderabad)

Recognised Under Section 2(f) & 12(B) of The UGC act, 1956

Dundigal, Quthbullapur (M), Hyderabad-500043.



C++ Programming Lab Manual

Subject Code: CS309PC
Regulation: R16- JNTUH
Class: B. Tech II-I Semester

Prepared by
M. Jagadeesh
Assistant Professor

**DEPARTMENT OF COMPUTER SCIENCE
AND ENGINEERING**

CERTIFICATE

This is to certify that this manual is a bonafide record of practical work in the **C++ Programming** in **First Semester of II year B.Tech (CSE) programme** during the academic year **2019-20**. This book is prepared by **Mr.M.Jagadeesh (Asst.Professor)**, Department of Computer Science and Engineering.

INDEX

S.No	Content	Page No:
	Preface	
	Acknowledgement	
	General Instructions	
	Safety Precautions	
	Institute Vision and Mission	
	Department Vision, Mission, Programme Educational Objectives and Specific Outcomes	
	Programme Outcomes	
	Course Structure, Objectives & Outcomes	
	Experiments Learning Outcomes	
1	Write a C++ program to display Names, Roll No., and grade of 3 students who have appeared in the examination. Declare the class of name, roll no., and grade. Create an array of class objects. Read and display the contents of the array.	
2	Write a C++ program to declare struct. Initialize and display contents of member	
3	Write a C++ program to declare a class . Declare pointer to class. Initialize and display the contents of the class members	
4	Given that EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary, and print data members.	
5	Write a C++ program to read the data of N employee and compute the net salary of each employee (DA=52% of basic and IT=30% of gross salary).	
6	Write a C++ to illustrate the concepts of console I/O operations	
7	Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.	
8	Write a C++ program to allocate memory using new operator.	
9	Write a C++ program to create multilevel inheritance	
10	Write a C++ program to create an array of pointers. Invoke functions using array objects	
11	Write a C++ program to use pointer for both base and derived classes and call the member function. Use virtual keyword	

PREFACE

This book “C++ Programming” lab manual is intended to teach the design and analysis of basic object oriented programming concepts and their implementation in an object-oriented programming language C++. Readers of this book need only be familiar with the basic syntax of C++ and similar languages. The “C++ Programming” is increasingly becoming the default choice of the IT industry especially industries involved in software development at system level. Therefore, for proper development of “C++ Programming” skills among the students this practical manual has been prepared. The manual contains the exercise programs and their solution for easy & quick understanding of the students. I hope that this practical manual will be helpful for students of Computer Science & Engineering for understanding the subject from the point of view of applied aspects. There is always scope for improvement in the manual. I would appreciate to receive valuable suggestions from readers and users for future use.

By,

M.Jagadeesh

ACKNOWLEDGEMENT

It was really a good experience, working with *C++ Programming* lab. First we would like to thank Mr. K.Abdul Basith, Assoc. Professor, HOD of Department of Computer Science and Engineering, Marri Laxman Reddy Institute of technology & Management for his concern and giving the technical support in preparing the document.

We are deeply indebted and gratefully acknowledge the constant support and valuable patronage of Dr. R. Kotaih, Director, Marri Laxman Reddy Institute of technology & Management for giving us this wonderful opportunity for preparing the *C++ Programming* laboratory manual.

We express our hearty thanks to Dr. K.Venkateswara Reddy, Principal, Marri Laxman Reddy Institute of technology & Management, for timely corrections and scholarly guidance.

At last, but not the least I would like to thanks the entire CSE Department faculties those who had inspired and helped us to achieve our goal.

By,

M.Jagadeesh

GENERAL INSTRUCTIONS

1. Students are instructed to come to Data Structures through C++ laboratory on time. Late comers are not entertained in the lab.
2. Students should be punctual to the lab. If not, the conducted experiments will not be repeated.
3. Students are expected to come prepared at home with the experiments which are going to be performed.
4. Students are instructed to display their identity cards before entering into the lab.
5. Students are instructed not to bring mobile phones to the lab.
6. Any damage/loss of system parts like keyboard, mouse during the lab session, it is student's responsibility and penalty or fine will be collected from the student.
7. Students should update the records and lab observation books session wise. Before leaving the lab the student should get his lab observation book signed by the faculty.
8. Students should submit the lab records by the next lab to the concerned faculty members in the staffroom for their correction and return.
9. Students should not move around the lab during the lab session.
10. If any emergency arises, the student should take the permission from faculty member concerned in written format.
11. The faculty members may suspend any student from the lab session on disciplinary grounds.
12. Never copy the output from other students. Write down your own outputs.

INSTITUTION VISION AND MISSION

VISION

To be as an ideal academic institution by graduating talented engineers to be ethically strong, competent with quality research and technologies

MISSION

To fulfill the promised vision through the following strategic characteristics and aspirations:

- Utilize rigorous educational experiences to produce talented engineers
- Create an atmosphere that facilitates the success of students
- Programs that integrate global awareness, communication skills and Leadership qualities
- Education and Research partnership with institutions and industries to prepare the students for interdisciplinary research

DEPARTMENT VISION, MISSION , PROGRAMME EDUCATIONAL OBJECTIVES AND SPECIFIC OUTCOMES

VISION

To empower the students to be technologically adept, innovative, self-motivated and responsible global citizen possessing human values and contribute significantly towards high quality technical education with ever changing world

MISSION

1. To offer high-quality education in the computing fields by providing an environment where the knowledge is gained and applied to participate in research, for both students and faculty.
2. To develop the problem solving skills in the students to be ready to deal with cutting edge technologies of the industry.
3. To make the students and faculty excel in their professional fields by inculcating the communication skills, leadership skills, team building skills with the organization of various co-curricular and extra-curricular programmes.
4. To provide the students with theoretical and applied knowledge, and adopt an education approach that promotes lifelong learning and ethical growth.

PROGRAMME EDUCATIONAL OBJECTIVES

The Programme Educational Objectives (PEOs) that are formulated for the civil engineering programme are listed below;

PEO1: Establish a successful professional career in industry, government or academia.

PEO2: Gain multidisciplinary knowledge providing a sustainable competitive edge in higher studies or Research.

PEO3: Promote design, analyze, and exhibit of products, through strong communication, leadership and ethical skills, to succeed an entrepreneurial.

PROGRAM SPECIFIC OUUTCOMES

PSO1: Applications of Computing: Ability to use knowledge in various domains to provide solution to new ideas and innovations.

PSO2: Programming Skills: Identify required data structures, design suitable algorithms, develop and maintain software for real world problems.

PROGRAMME OUT COMES

The Program Outcomes (POs) of the department are defined in a way that the Graduate Attributes are included, which can be seen in the Program Outcomes (POs) defined. The Program Outcomes (POs) of the department are as stated below:

- a : An ability to apply knowledge of Science, Mathematics, Engineering & Computing fundamentals for the solutions of Complex Engineering problems.
- b : An ability to identify, formulates, research literature and analyze complex engineering problems using first principles of mathematics and engineering sciences.
- c : An ability to design solutions to complex process or program to meet desired needs.
- d : Ability to use research-based knowledge and research methods including design of experiments to provide valid conclusions.
- e : An ability to use appropriate techniques, skills and tools necessary for computing practice.
- f : Ability to apply reasoning informed by the contextual knowledge to assess social issues, consequences & responsibilities relevant to the professional engineering practice.
- g : Ability to understand the impact of engineering solutions in a global, economic, environmental, and societal context with sustainability.
- h : An understanding of professional, ethical, Social issues and responsibilities.
- i : An ability to function as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.
- j : An ability to communicate effectively on complex engineering activities within the engineering community.
- k : Ability to demonstrate and understanding of the engineering and management principles as a member and leader in a team.
- l : Ability to engage in independent and lifelong learning in the context of technological change.

COURSE STRUCTURE, OBJECTIVES & OUTCOMES

Laboratory subjects – Internal and external evaluation – Details of marks

Data Structures through C++ lab will have a continuous evaluation during 3rd semester for 25 sessional marks and 75 end semester examination marks.

Out of the 25 marks for internal evaluation, day-to-day work in the laboratory shall be evaluated for 15 marks and internal practical examination shall be evaluated for 10 marks conducted by the laboratory teacher concerned.

The end examination will be evaluated for a maximum of 75 marks. The end semester examination shall be conducted with an external examiner and internal examiner. The external examiner shall be appointed by the principal / Chief Controller of examinations

Course Objectives:

- To write and execute programs in C++ to solve problems using data structures such as arrays, linked lists, stacks, queues, trees, graphs, hash tables and search trees.
- To learn and write C++ programs to implement various sorting and searching algorithms

Course Outcomes:

- Able to identify the appropriate data structures and algorithms for solving real time problems.
- Able to implement various kinds of searching and sorting techniques.
- Able to implement data structures such as stacks, queues, Search trees, and hash tables to solve various computing problems.

1. Write a C++ program to display Names, Roll No., and grade of 3 students who have appeared in the examination. Declare the class of name, roll no., and grade. Create an array of class objects. Read and display the contents of the array.

Algorithm:

1. Define the class consisting the members of name, rollno and grade of the student.
2. Read the student name, roll number and grade.
3. Print the student name, roll number and grade.

Source Code:

```
#include<stdio.h>
#include<conio.h>
#include<iostream.h>
void main()
{
int k=0;
Class stud
{
public: char name[12];
      int rollno;
      char grade[2];
};
class stud st[3];
while(k<3)
{ clrscr();
gotoxy(2,4);
cout<<"Name";
gotoxy(17,4);
cin>>st[k].name;
gotoxy(2,5);
cout<<"Roll Number";
gotoxy(17,5);
cin>>st[k].rollno;
cout<<"Grade";
gotoxy(17,6);
cin>>st[k].grade;
st[k].grade[1]='\0';
puts("press any key..");
getch();
k++;
}

k=0;
clrscr();
cout<<"\n Name\tRoll No.\t Grade\n";
while(k<3)
```

```
{  
cout<<st[k].name<<"\t"<<st[k].rollno<<"\t"<<st[k].grade<<"\n";  
k++;  
}  
}
```

Output:

Name	RollNo	Grade
Balaji	50	A
Manoj	51	B
Sanjay	55	C

2 Write a C++ program to declare struct. Initialize and display contents of member variables.

Algorithm:

1. Define the structure consisting of members name , roll number and marks of the student
2. Read the student name, roll number and marks of the student.
3. Print the student name, roll number and marks of the student.

Source Code:

```
#include <iostream>
using namespace std;
struct student
{
    char name[50];
    int roll;
    float marks;
};
int main()
{
    student s;
    cout << "Enter information," << endl;
    cout << "Enter name: ";
    cin >> s.name;
    cout << "Enter roll number: ";
    cin >> s.roll;
    cout << "Enter marks: ";
    cin >> s.marks;
    cout << "\nDisplaying Information," << endl;
    cout << "Name: " << s.name << endl;
    cout << "Roll: " << s.roll << endl;
    cout << "Marks: " << s.marks << endl;
    return 0;
}
```

Output

```
Enter information,
Enter name: Bill
Enter roll number: 4
Enter marks: 55.6
Displaying Information,
Name: Bill
Roll: 4
Marks: 55.6
```

3. Write a C++ program to declare a class . Declare pointer to class. Initialize and display the contents of the class members

Algorithm:

1. Define a class with one class member.
2. Declare the pointer object to the class in the main function.
3. In main function, with the pointer object we can access the class member.

Source Code

```
class Simple
{
    public:
    int a=5;
};
int main()
{
    Simple obj;
    Simple* ptr; // Pointer of class type
    ptr = &obj;

    cout << obj.a;
    cout << ptr->a; // Accessing member with pointer
}
```

Output:

5
5

4. Given that EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary, and print data members.

Source Code:

```
#include<iostream>
using namespace std;
class employee
{
    Int empno;
    Char empname[50];
    Int basic;
    Int da;
    Int it;
    Int gs;
    Int ns;
    Public:
    Void getdata()
    {
        Cout<<"enter the employee number";
        Cin>>empno;
        Cout<<"enter the employee name";
        Cin>>empname;
        Cout<<"enter the basic salary,da,it";
        Cin>>basic>>da>>it;
        Gs=da+it;
        Ns=gs+basic
    }
    Void putdata()
    {
        Cout<<"\nEmployee number"<<empno;
        Cout<<"\nEmployee name"<<empname;
        Cout<<"\nbasic\t da\t it\n"<<basic<<da<<it;
        Cout<<"\n gross salary\n net salary"<<gs<<ns;
    }
};
Void main()
{
    Employee e;
    e.getdata();
    e.putdata();
}
```

Output:

```
Enter the employee number 101
Enter the employee name Rakesh
Enter the basic salary , da, it
15000
500
```

600

Employee number101

Employee name Rakesh

Basic da ta 15000 500 600

Gross salary1100 netsalary 16100

5. Write a C++ program to read the data of N employee and compute the net salary of each employee (DA=52% of basic and IT=30% of gross salary)

Source Code

```
#include<iostream>
using namespace std;
class employee
{
    Int empno;
    char empname[50];
    int basic;
    int da;
    int it;
    int gs;
    int ns;
public:
    void getdata()
    {
        Cout<<"enter the employee number";
        Cin>>empno;
        Cout<<"enter the employee name";
        Cin>>empname;
        Cout<<"enter the basic salary,da,it";
        Cin>>basic>>da>>it;
        Gs=basic+da+it;
        Ns=
    }
    Void putdata()
    {
        Cout<<"\nEmployee number"<<empno;
        Cout<<"\nEmployee name"<<empname;
        Cout<<"\nbasic\tda\tit\n"<<basic<<da<<it;
        Cout<<"\n gross salary\n net salary"<<gs<<ns;
    }
};
Void main()
{
    Employee e[20];
    Int n,i;
    Cout<<"enter the number of employees";
    Cin>>n;
    For(i=0;i<n;i++)
    {
        Cout<<"enter the employee <<i<< details<<"\n";
        e[i].getdata();
        e[i].putdata();
    }
}
```

```
}
```

6. Write a C++ to illustrate the concepts of console I/O operations

a) get()

```
#include<iostream>
using namespace std;
int main()
{
    char c=cin.get();
    cout<<c<<endl;
    return 0;
}
```

Output

```
I
I
```

b) put()

```
#include<iostream>
using namespace std;
int main()
{
    char c=cin.get();
    cout.put(c); //Here it prints the value of variable c;
    cout.put('c'); //Here it prints the character 'c';
    return 0;
}
```

Output

```
I
Ic
```

C) getline(char *buffer,int size)

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Enter name :";
    char c[10];
    cin.getline(c,10); //It takes 10 characters as input;
    cout<<c<<endl;
    return 0;
}
```

Output

```
Enter name :Divyanshu
Divyanshu
```

D) write(char * buffer, int n)

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Enter name : ";
    char c[10];
    cin.getline(c,10); //It takes 10 charcters as input;
    cout.write(c,9); //It reads only 9 character from buffer c;
    return 0;
}
```

Output

Enter name : Divyanshux
Divyanshu

E) cin

```
#include<iostream>
using namespace std;
int main()
{
    int num;
    char ch;
    string str;
    cout<<"Enter Number"<<endl;
    cin>>num; //Inputs a variable;
    cout<<"Enter Character"<<endl;
    cin>>ch; //Inputs a character;
    cout<<"Enter String"<<endl;
    cin>>str; //Inputs a string;
    return 0;
}
```

Output

Enter Number
07
Enter Character
h
Enter String
Deepak

F) cout

```
#include<iostream>
using namespace std;
int main()
{
    int num=100;
    char ch='X';
    string str="Deepak";
    cout<<"Number is "<<num<<endl; //Prints value of variable;
    cout<<"Character is "<<ch<<endl; //Prints character;
    cout<<"String is "<<str<<endl; //Prints string;
```

```
    return 0;  
}
```

Output

Number is 100

Character is X

String is Deepak

7. Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.

Source Code:

```
#include<iostream>
int n = 12; //global variable
int main() {
    int n = 13; //local variable
    cout << ::n << endl; //print global variable:12
    cout << n << endl; //print the local variable:13
}
```

Output:

12

13

8. Write a C++ program to allocate memory using new operator.

Source Code:

```
#include <iostream>
using namespace std;
int main()
{
    int n, *pointer, c;
    cout << "Input an integer\n";
    cin >> n;
    pointer = new int[n];
    cout << "Input " << n << " integers\n";

    for (c = 0; c < n; c++)
        cin >> pointer[c];

    cout << "Elements entered by you are\n";

    for (c = 0; c < n; c++)
        cout << pointer[c] << endl;

    delete[] pointer;

    return 0;
}
```

Output

```
Input an integer
2
Input 2 integers
456
-98
Elements entered by you are
456
-98
```

9. Write a C++ program to create multilevel inheritance

Source Code

```
#include <iostream>
#include <conio.h>
using namespace std;

class person
{
    char name[100],gender[10];
    int age;
public:
    void getdata()
    {
        cout<<"Name: ";
        fflush(stdin); /*clears input stream*/
        gets(name);
        cout<<"Age: ";
        cin>>age;
        cout<<"Gender: ";
        cin>>gender;
    }
    void display()
    {
        cout<<"Name: "<<name<<endl;
        cout<<"Age: "<<age<<endl;
        cout<<"Gender: "<<gender<<endl;
    }
};

class employee: public person
{
    char company[100];
    float salary;
public:
    void getdata()
    {
        person::getdata();
        cout<<"Name of Company: ";
        fflush(stdin);
        gets(company);
        cout<<"Salary: Rs.";
        cin>>salary;
    }
    void display()
    {
        person::display();
        cout<<"Name of Company: "<<company<<endl;
        cout<<"Salary: Rs."<<salary<<endl;
    }
}
```

```

};

class programmer: public employee
{
    int number;
public:
    void getdata()
    {
        employee::getdata();
        cout<<"Number of programming language known: ";
        cin>>number;
    }
    void display()
    {
        employee::display();
        cout<<"Number of programming language known: "<<number;
    }
};

int main()
{
    programmer p;
    cout<<"Enter data"<<endl;
    p.getdata();
    cout<<endl<<"Displaying data"<<endl;
    p.display();
    getch();
    return 0;
}

```

Output

```

Enter data
Name: Karl Lens
Age: 31
Gender: Male
Name of Company: Dynamic Info
Salary: $21000
Number of programming language known: 4

```

```

Displaying data
Name: Karl Lens
Age: 31
Gender: Male
Name of Company: Dynamic Info
Salary: $21000
Number of programming language known: 4

```


10. Write a C++ program to create an array of pointers. Invoke functions using array objects

```
#include <iostream>
using namespace std;
const int MAX = 3;
int main () {
    int var[MAX] = {10, 100, 200};
    int *ptr[MAX];
    for (int i = 0; i < MAX; i++) {
        ptr[i] = &var[i]; // assign the address of integer.
    }
    for (int i = 0; i < MAX; i++) {
        cout << "Value of var[" << i << "] = ";
        cout << *ptr[i] << endl;
    }

    return 0;
}
```

Output:

```
Value of var[0] = 10
Value of var[1] = 100
Value of var[2] = 200
```

11. Write a C++ program to use pointer for both base and derived classes and call the member function. Use virtual keyword

Source Code

```
class Base
{
    public:
    virtual void show()
    {
        cout << "Base class\n";
    }
};
class Derived:public Base
{
    public:
    void show()
    {
        cout << "Derived Class";
    }
}

int main()
{
    Base* b;    //Base class pointer
    Derived d; //Derived class object
    b = &d;
    b->show(); //Late Binding Ocuurs
}
```

Output:

Derived class