



Estd. 2001

Sri Indu

College of Engineering & Technology

UGC Autonomous Institution

Recognized under 2(f) & 12(B) of UGC Act 1956,

NAAC, Approved by AICTE &

Permanently Affiliated to JNTUH



NAAC

NATIONAL ASSESSMENT AND
ACCREDITATION COUNCIL



HANDOUT

II Year I Semester

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ACADEMIC YEAR 2022-2023



SRIINDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

(Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi and Accredited by NBA, NAAC)
Sheriguda Village, Ibrahimpatnam Mandal, Ranga Reddy Dist. – 501 510

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB HANDOUT- INDEX

S. No	Contents
1	Vision, Mission, PEOs, PSOs, POs & Cos
2	Institution Academic Calendar
3	Department Academic Calendar
4	Lab Manual



SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

**(Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi and Accredited by NBA, NAAC)
Sheriguda Village, Ibrahimpatnam Mandal, Ranga Reddy Dist. – 501 510**

I

INSTITUTION VISION

To be a premier institution in engineering & technology and management with competence, values and social consciousness.

INSTITUTION MISSION

IM₁: Provide high quality academic programmes, training activities and research facilities.

IM₂: Promote continuous industry-institute interaction for employability, entrepreneurship, leadership and research aptitude among stakeholders.

IM₃: Contribute to the economic and technological development of the region, state and nation.



SRIINDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

**(Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi and Accredited by NBA, NAAC)
Sheriguda Village, Ibrahimpatnam Mandal, Ranga Reddy Dist. – 501 510**

VISION OF THE DEPARTMENT

To be a technologically adaptive centre for computing by grooming the students as top notch professionals.

MISSION OF THE DEPARTMENT

DM₁: To offer quality education in computing.

DM₂: To provide an environment that enables overall development of all the stakeholders.

DM₃: To impart training on emerging technologies like Data Analytics, Artificial Intelligence and Internet Of Things.

DM₄: To encourage participation of stakeholders in research and development

Program Educational Objectives(PEO's)

PEO1	Higher Studies : Graduates with an ability to apply knowledge of Basic Sciences and programming skills in their career and higher education.
PEO2	Lifelong Learning: Graduates with an ability to adopt new technologies for ever changing IT industry needs through Self-Study, Critical thinking and Problem solving skills.
PEO3	Professional Skills : Graduates will be ready to work in projects related to complex problems involving multidisciplinary projects with effective analytical skills
PEO4	Engineering citizenship: Graduates with an ability to communicate well and exhibit social,technical and ethical responsibility in process or product.

Program Specific Outcomes(PSO's)

PSO1	Software Development: To apply the knowledge of Software Engineering, Data Communication, Web Technology and Operating Systems for building IOT and Cloud Computing applications.
PSO2	Industrial Skills Ability: Design, develop and test software systems for world-wide network of computers to provide solutions to real world problems.
PSO3	Project Implementation: Analyze and recommend the appropriate IT infrastructure required for the implementation of a project.

Program Outcomes(PO's)

PO1	Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	Problem Analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3	Design / Development of Solutions: 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.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: 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.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

COURSE OUTCOMES (CO's)

Academic Year: 2022-23

Class: II YEAR-I SEM.

Course Name: C++ Programming (R20CSE21L3)

At the end of the course, the student will be able to

Course Outcomes (COs)	
C21L3.1	Develop applications for a range of problems using object-oriented programming(Create)
C21L3.2	Demonstrate the implementation of constructors, destructors and operatoroverloading. (Apply)
C21L3.3	Apply virtual and pure virtual function & complex program situations(Apply)
C21L3.4	Apply fundamental algorithmic problems including type casting, inheritance, andpolymorphism. (Apply)
C21L3.5	Explain generic programming, templates, file handling. (Understand)
C21L3.6	Handle exceptions in programming (Analyze)

Mapping of Course Outcomes(CO's) with PO's:

CO	PO											
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
C21L3.1	3	2	3	3	1	1	-	-	-	-	3	1
C21L3.2	3	3	3	3	-	-	-	-	-	-	-	1
C21L3.3	3	3	3	3	-	-	-	-	1	-	1	1
C21L3.4	3	2	3	2	-	1	-	-	-	-	-	1
C21L3.5	3	3	3	3	-	-	-	-	1	-	-	2
C21L3.6	3	3	3	3	-	-	-	-	-	-	-	1
	3	2.6	3	2.6	-	1	-	-	1	-	3	1.5

3: High 2: Medium 1: Low



LR.NO.SICET/AUTO/DAE/BR-20/ACADEMIC-CA_L/520/2022

DATE: 05.09.2022

II B.TECH ACADEMIC CALENDAR
ACADEMIC YEAR : 2022-2023

Dr. G. SURESH,
Principal,
To,
All the HODs.
Sir,

Red

Sub: SICET (Autonomous) - Academic & Evaluation - Academic Calendar for
II B.Tech - I & II Semester for the academic year **2022-23** – Reg.

The approved Academic Calendar for **II B.Tech – I & II Semester** for the academic
Year **2022-23** is given below:

ACADEMIC CALENDAR - II B.TECH - I & II SEMESTER
ADMITTED BATCH – (2021 – 2022) of BR-20 Regulation.

I SEMESTER

Commencement of I Sem class work	26.09.2022	
I Spell of Instructions (Including Dussehra Holidays).	26.09.2022 - 26.11.2022	9 Weeks
Dussehra Holidays.	03.10.2022 - 08.10.2022	1 Week
I Mid Examinations for II B.Tech I Sem Students.	28.11.2022 - 30.11.2022	3 Days
II Spell of Instructions.	01.12.2022 - 28.01.2023	8 Weeks 3 Days
Sankranti Holidays.	13.01.2023 - 16.01.2023	4 Days
II Mid Examinations for II B.Tech I Sem Students.	30.01.2023 - 01.02.2023	3 Days
Preparation Holidays, Practical Lab Examinations and Remedial Mid Test (RMT).	02.02.2023 - 11.02.2023	10 Days
II B.Tech I Semester End Examinations (Main) and Supplementary Examinations.	13.02.2023 - 25.02.2023	2 Weeks
Commencement of Class-Work for II B.Tech - II Semester 27.02.2023 (Monday).		

II SEMESTER

Commencement of II Sem class work.	27.02.2023	
I Spell of Instructions.	27.02.2023 - 22.04.2023	8 Weeks
I Mid Examinations for II B.Tech. II Sem. Students.	24.04.2023 - 26.04.2023	3 Days
II Spell of Instructions.	27.04.2023 - 05.07.2023	10 Weeks
Summer Vacation.	15.05.2023 - 27.05.2023	2 Weeks
II Mid Examinations for II B.Tech. II Sem. Students.	06.07.2023 - 08.07.2023	3 Days
Preparation Holidays, Practical Lab Examinations Remedial Mid Test (RMT).	10.07.2023 - 19.07.2023	10 Days
II B.Tech II Semester End Examinations (Main) and Supplementary Examinations.	20.07.2023 - 02.08.2023	2 Weeks
Commencement of Class-Work for III B.Tech - I Semester 07.08.2023 (Monday).		

16:01

Subs

Red

8000

SRI INDU COLLEGE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
DEPARTMENT Academic CALENDAR – 2022-2023 (SEMESTER-1)

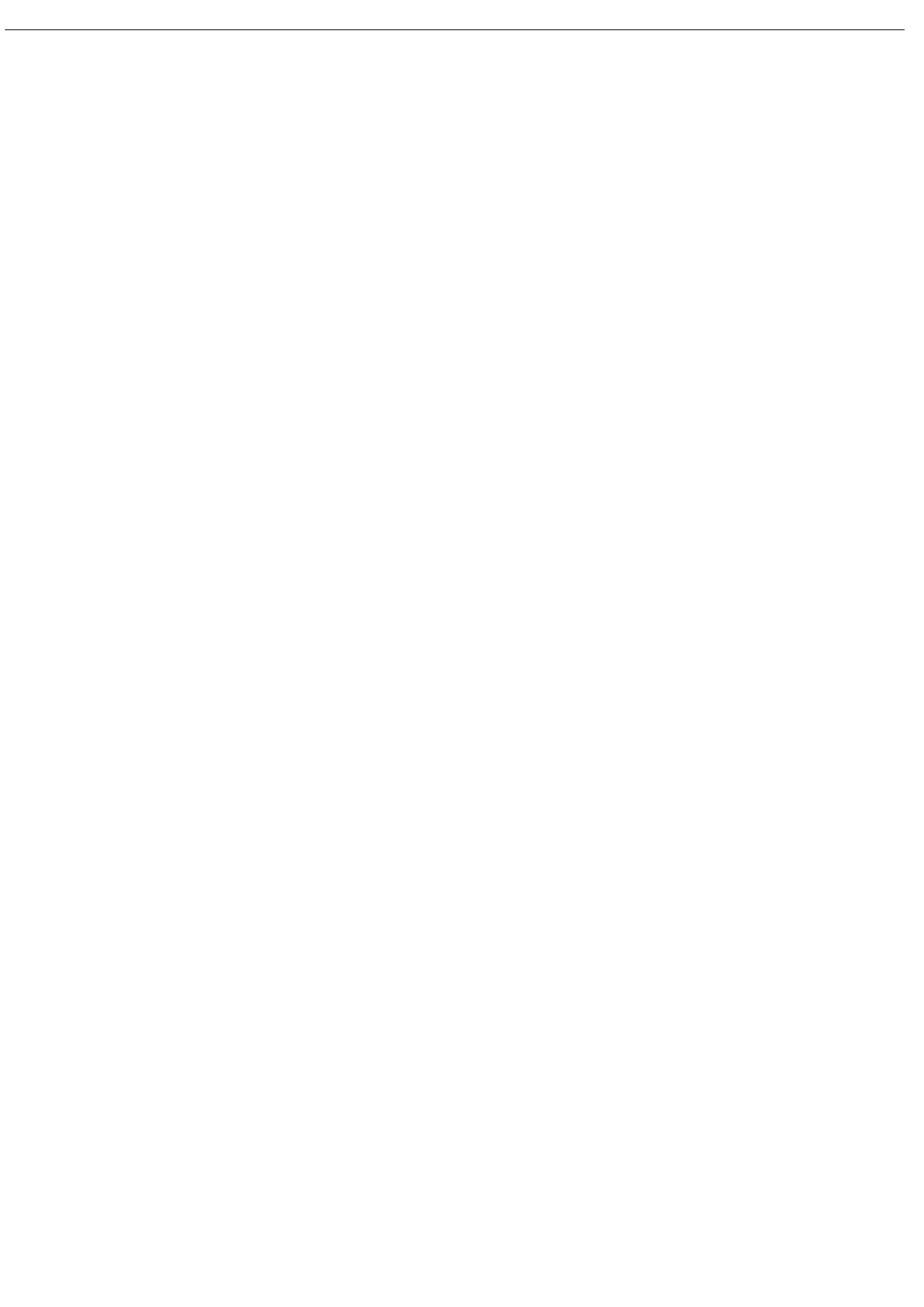
DAYS							
SUNDAY			NOVEMBER '22				
MONDAY			1		DECEMBER '22		FEBRUARY'23
TUESDAY	SEPTEMBER'22		2				1 MID EXAM-I
WEDNESDAY	2		3		1		2 PRACTICAL EXAM
THURSDAY	1	OCTOBER '22	4		2		3 PRACTICAL EXAM
FRIDAY	2	Bathukamma Celebrations	5		3	JANUARY'23	4 PRACTICAL EXAM
SATURDAY	3	Gandhi Jayanti/ HOLIDAY	6	HOLIDAY	4 HOLIDAY	NEW YEAR/ HOLIDAY	5 HOLIDAY
SUNDAY	4	HOLIDAY	7		5		6 PRACTICAL EXAM
MONDAY	5	DASARA HOLIDAYS	8	GURUNANAK JAYANTHI	6		7 PRACTICAL EXAM
TUESDAY	6	DASARA HOLIDAYS	9		7		8 PRACTICAL EXAM
WEDNESDAY	7	DASARA HOLIDAYS	10		8		9 PRACTICAL EXAM
THURSDAY	8	DASARA HOLIDAYS	11		9		10 PRACTICAL EXAM
FRIDAY	9	Ganesh Nimajanam	12		10		11 PRACTICAL EXAM
SATURDAY	10	HOLIDAY	13	HOLIDAY	11 HOLIDAY	HOLIDAY	12 HOLIDAY
SUNDAY	11	HOLIDAY	14		12		13 END EXAM
MONDAY	12		15		13		14 END EXAM
TUESDAY	13		16		14		15 END EXAM
WEDNESDAY	14		17		15		16 END EXAM
THURSDAY	15		18		16	BHOGI	17 END EXAM
FRIDAY	16		19		17	SANKRANTHI	18 END EXAM

SATURDAY	17	Telangana vimochana dinostavam	18	HOLIDAY	20	HOLIDAY	18	HOLIDAY	19	HOLIDAY	19	HOLIDAY
SUNDAY	18	HOLIDAY	17		21		19		18	HOLIDAY	20	END EXAM
MONDAY	19		18		22		20		17		21	END EXAM
TUESDAY	20		19		23		21		18		22	END EXAM
WEDNESDAY	21		20		24		22		19		23	END EXAM
THURSDAY	22		21		25		23		20		24	END EXAM
FRIDAY	23		22		26		24		21		25	END EXAM
SATURDAY	24		23	HOLIDAY	27	HOLIDAY	25	CHRISTMAS HOLIDAY	22	HOLIDAY	26	
SUNDAY	25	HOLIDAY	24		28	MID EXAM-I	26	BOXING DAY HOLIDAY	23		27	
MONDAY	26	Commenceme of Classes (I Yr)	25	DIWALI	29	MID EXAM-I	27		24		28	
TUESDAY	27		26		30	MID EXAM-I	28		25		29	
WEDNESDAY	28		27				29		26	Republic Day		
THURSDAY	29		28				30		27			
FRIDAY	30		29				31		28			
SATURDAY			30						29	HOLIDAY		
SUNDAY			31						30	MID EXAM-II		
MONDAY									31	MID EXAM-II		

CALENDAR INCHARGE

HOD/CSE
DEAN

PRINCIPAL



SRI INDU COLLEGE OF ENGINEERING & TECHNOLOGY

(An Autonomous Institution under UGC, New Delhi)

B.Tech. - II Year – I Semester

L	T	P	C
0	0	2	1

(R20CSE21L3) C++ Programming Lab

Course Objectives


- Introduces object-oriented programming concepts using the C++ language.
- Introduces the principles of data abstraction, inheritance and polymorphism;
- Introduces the principles of virtual functions and polymorphism
- Introduces handling formatted I/O and unformatted I/O
- Introduces exception handling

Course Outcomes

- Ability to develop applications for a range of problems using object-oriented programming Techniques
- Demonstrate the concepts of inheritance and polymorphism.
- Difference between function overloading and function overriding
- Explain exception handling in object-oriented programs.
- Use template classes and the STL library in C++.
- write object-oriented programs of moderate complexity in C++
- Analyze operators in C++.

LIST OF EXPERIMENTS

1. Write a C++ Program to display Names, Roll No., and grades 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 variables.
3. Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.
4. Given that an 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 Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the 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. (Hint: Classes A1, A2, A3)
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.

	SRI INDU COLLEGE OF ENGG & TECH		Prepared on Rev1:
	LAB PLAN (Regulation :R20)		Page: 1 of 4
	Department of Computer Science and Engineering		
	Sub. Code & Title	(R20CSE21L3) Object Oriented Programming using C++ LAB	
Academic Year: 2022-23	Year/Sem./Section	II/I/A&B&C&D	
Faculty Name & Designation	1. DR.N.SADASHIVAM (PROFESSOR), 2. A.SUDHEER ASST.PROF, 3. G.AKILA ASST.PROF, 4. K.KRISHNA ASST.PROF		

Lab Plan

2022-23 II Year –I Semester CSE

S No	Topics	No. of weeks
1.	Write a C++ Program to display Names, Roll No., and grades 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.	1
2.	Write a C++ program to declare Struct. Initialize and display contents of member variables.	1
3.	Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.	1
4.	Given that an EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary and print data members.	1
5.	Write a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross	1

	salary).	
6.	Write a C++ to illustrate the concepts of console I/O operations.	1
7.	Write a C++ program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.	1
8.	Write a C++ program to allocate memory using new operator.	1
9.	Write a C++ program to create multilevel inheritance. (Hint: Classes A1, A2, A3)	1
10.	Write a C++ program to create an array of pointers. Invoke functions using array objects.	1
11.	Write a C++ program to use pointer for both base and derived classes and call the memberfunction. Use Virtual keyword.	1

Object Oriented Programming using C++ Lab Manual

(Subject Code: R20CSE21L3)

Computer Science & Engineering

Lab Manual

C++ Programming

1. Write a C++ Program to display Names, Roll No., and grades 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 variables.
3. Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.
4. Given that an 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 Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the 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. (Hint: Classes A1, A2, A3)
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 their member function. Use Virtual keyword

PROGRAMS

Week 1.

Aim: Write a C++ Program to display Names, Roll No., and grades 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.

Source Code:

```
#include <iostream> using
namespace std;

class student {

    public:
    charname[50];
    int roll;
    chgrade;
} s[3];

int main() {
    cout <<"Enter information of students:" << endl;

    // storing information
    for(int i = 0; i <3; ++i) {
        s[i].roll =i+1;
        cout <<"For roll number" <<s[i].roll <<" ," <<endl;

        cout <<"Enter name: ";
        cin >>s[i].name;

        cout <<"Enter Grade: "; cin
        >>s[i].grade;

        cout << endl;
    }

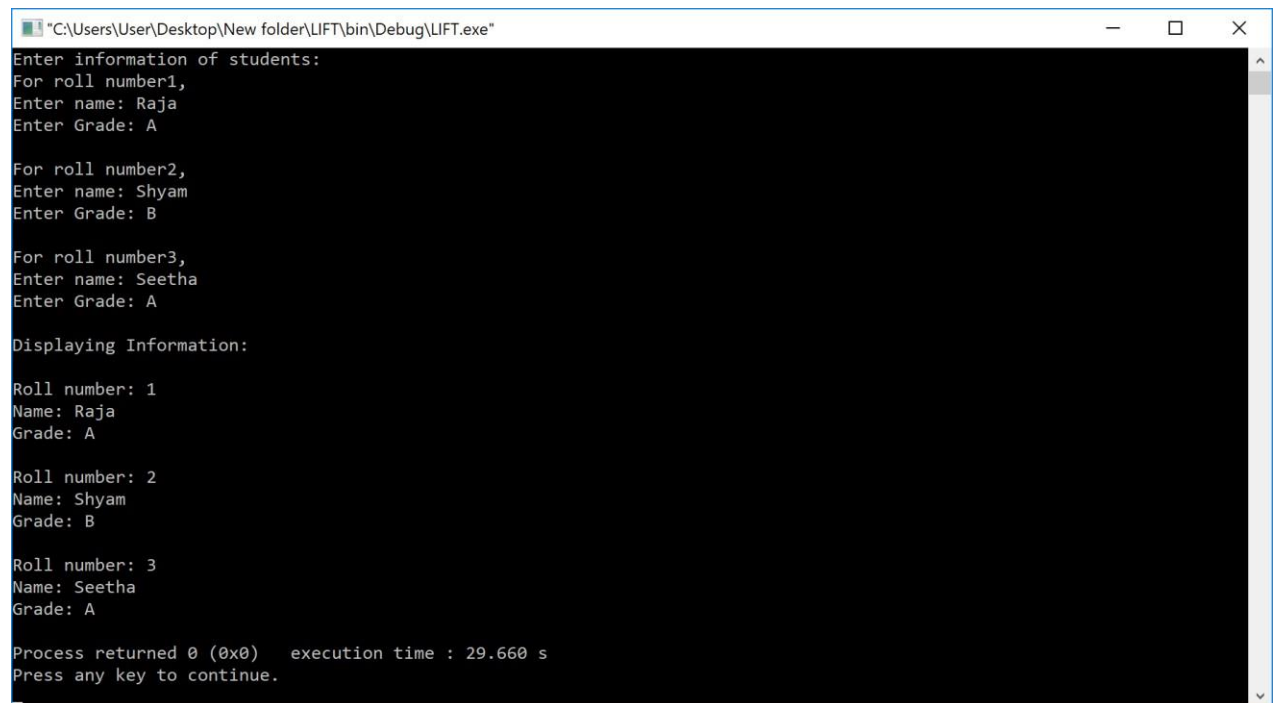
    cout <<"Displaying Information: " << endl;

    // Displaying information
    for(int i = 0; i <3; ++i) {
        cout <<"\nRoll number: " <<i+1 <<endl;
        cout <<"Name: " <<s[i].name <<endl; cout
        <<"Grade: " <<s[i].grade <<endl;
    }
}
```



```
    return 0;  
}
```

Output:



```
*C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe
Enter information of students:
For roll number1,
Enter name: Raja
Enter Grade: A

For roll number2,
Enter name: Shyam
Enter Grade: B

For roll number3,
Enter name: Seetha
Enter Grade: A

Displaying Information:

Roll number: 1
Name: Raja
Grade: A

Roll number: 2
Name: Shyam
Grade: B

Roll number: 3
Name: Seetha
Grade: A

Process returned 0 (0x0)   execution time : 29.660 s
Press any key to continue.
```

Week 2:

Aim: Write a C++ program to declare Struct. Initialize and display contents of member variables.

Source Code:

```
#include <iostream> using
namespace std;

struct Person
{
    char name[50];
    int age;
    float salary;
};

int main()
{
    Person p1;

    cout <<"Enter Full name: ";
    cin.get(p1.name,50);
    cout <<"Enter age: "; cin
    >> p1.age;
    cout <<"Enter salary: "; cin
    >>p1.salary;

    cout <<"\nDisplaying Information." <<endl;
    cout<<"Name:"<<p1.name<<endl;
    cout <<"Age: " << p1.age <<endl; cout
    <<"Salary: " <<p1.salary;

    return 0;
}
```

```
"C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe"
Enter Full name: Rajinikanth B
Enter age: 32
Enter salary: 35000

Displaying Information.
Name: Rajinikanth B
Age: 32
Salary: 35000
Process returned 0 (0x0)   execution time : 31.406 s
Press any key to continue.
```

Output:

Week 3:

Aim: Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member.

Sourcecode:

```
#include <iostream>

using namespace std;

class Data
{
    public:
    int a;
    void print()
    {
        cout << "\n" << "Value of 'a' is " << a << "\n";
    }
};

int main()
{
    Data d, *dp;
    dp=&d; // pointer to object

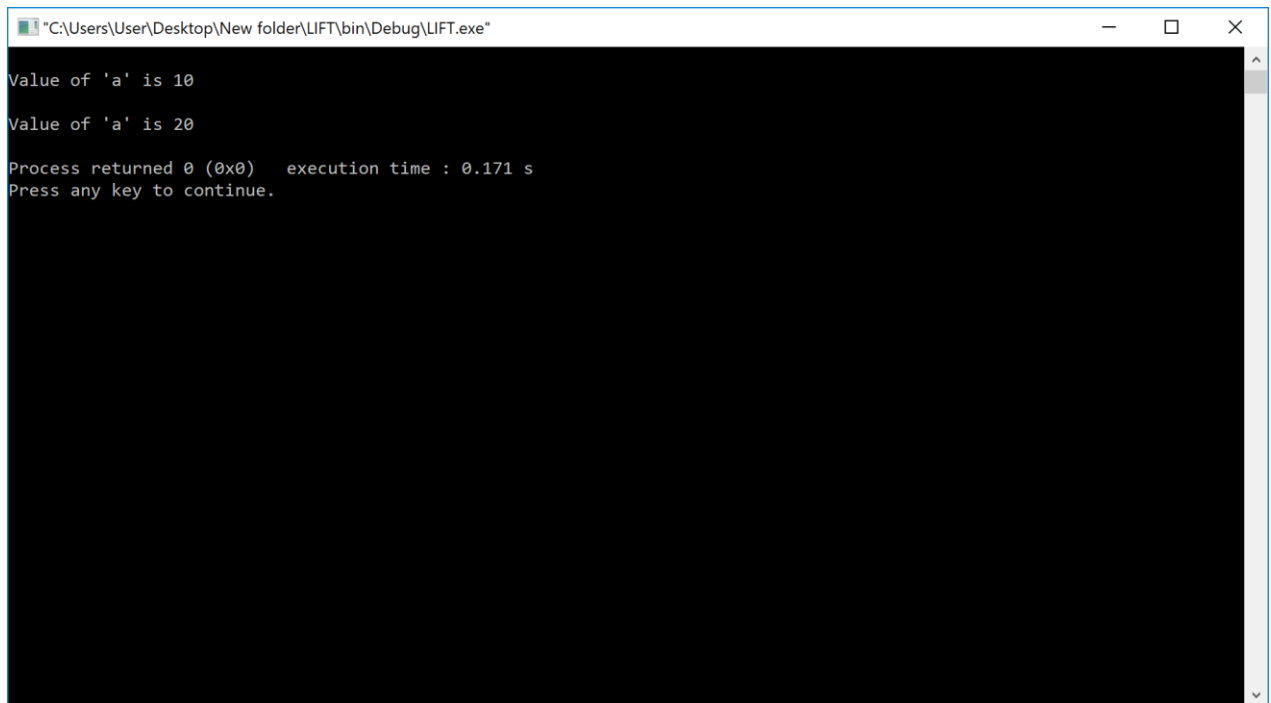
    int Data::*ptr=&Data::a; // pointer to data member 'a'

    d.*ptr=10;
    d.print();

    dp->*ptr=20;
    dp->print();

    return 0;
}
```


Output:



```
"C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe"
Value of 'a' is 10
Value of 'a' is 20
Process returned 0 (0x0)   execution time : 0.171 s
Press any key to continue.
```

Week 4:

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

Source code:

```
#include <iostream>
#include<conio.h>

using namespace std;

class employee
{
    int emp_number;
    char emp_name[20];
    float emp_basic;
    float sal;
    float emp_da;
    float emp_net_sal;
    float emp_it;

    public:
        void get_details(int i);
        void find_net_sal();
        void show_emp_details();
};

void employee :: get_details(int i) {
    cout<<"\nEnter employee " << i+1 <<" number: "; cin>>emp_number;
    cout<<"\nEnter employee " << i+1 <<" name: ";
    cin>>emp_name;
    cout<<"\nEnter employee " << i+1 <<" basic: "; cin>>emp_basic;
}

void employee :: show_emp_details() {
    cout<<"\n\n\nDetailsof : "<<emp_name;
    cout<<"\n\nEmployee number: "<<emp_number;
    cout<<"\nBasic salary : "<<emp_basic;
    cout<<"\nEmployee DA : "<<emp_da;
    cout<<"\nIncome Tax : "<<emp_it;
    cout<<"\nNet Salary : "<<emp_net_sal;
}

int main() {
    employee emp[10];
    int i,num;

    cout<<"\nEnter number of employee details: ";
```

```
cin>>num;

for(i=0;i<num;i++)
    emp[i].get_details(i);

for(i=0;i<num;i++)
    emp[i].show_emp_details();

    getch();
return 0;
}
```

Output:



```
C:\Users\User\Desktop\New folder\UFT\bin\Debug\UFT.exe
Enter number of employee details: 3
Enter employee 1 number: 111
Enter employee 1 name: Raghu
Enter employee 1 basic: 1000
Enter employee 2 number: 222
Enter employee 2 name: Santhu
Enter employee 2 basic: 2000
Enter employee 3 number: 333
Enter employee 3 name: Balu
Enter employee 3 basic: 1500

Details of   : Raghu
Employee number: 111
Basic salary   : 1000
Employee DA    : 0
Income Tax     : 1.24916e-038
Net Salary     : 4.61868e-042

Details of   : Santhu
Employee number: 222
Basic salary   : 2000
Employee DA    : 0
Income Tax     : 1.25289e-038
Net Salary     : 1.01014e-038

Details of   : Balu
Employee number: 333
Basic salary   : 1500
Employee DA    : 7.01928e-039
Income Tax     : 5.88411e-039
Net Salary     : 6.9739e-039
```

Week 5:

Aim: Write a C++ program to read the data of N employee and compute Net salary of each employee (DA=52% of Basic and Income Tax (IT) =30% of the gross salary).

Source code:

```
#include <iostream>
#include<conio.h>

using namespace std;

class employee {
    int emp_num;
    char emp_name[20];
    float emp_basic;
    float sal;
    float emp_da;
    float net_sal;
    float emp_it;

public:
    void get_details(int i);
    void find_net_sal();
    void show_emp_details();
};

void employee::get_details(int i) {
    cout<<"\nEnter employee "<<i+1<<" number: ";
    cin>>emp_num;
    cout<<"\nEnter employee "<<i+1<<" name: ";
    cin>>emp_name;
    cout<<"\nEnter employee "<<i+1<<" basic: "; cin>>emp_basic;
}

void employee::find_net_sal() {
    emp_da=0.52*emp_basic;
    emp_it=0.30*(emp_basic+emp_da);
    net_sal=(emp_basic+emp_da)-emp_it;
}

void employee::show_emp_details() {
    cout<<"\n\n\n*****EmployeeDetails*****\n";
    cout<<"\nDetail of : "<<emp_name;
    cout<<"\n\nEmployee number:
        "<<emp_num;cout<<"\nBa
        sics salary : "<<emp_basic;
    cout<<"\nEmployeeDA : "<<emp_da;
    cout<<"\nIncomeTax : "<<emp_it;
    cout<<"\nNetSalary : "<<net_sal;
}
```

```

int main() {
    employee emp[10];
    int i,num;

    cout<<"\nEnter number of employee details:";
    cin>>num;

    for(i=0;i<num;i++)
        emp[i].get_details(i);

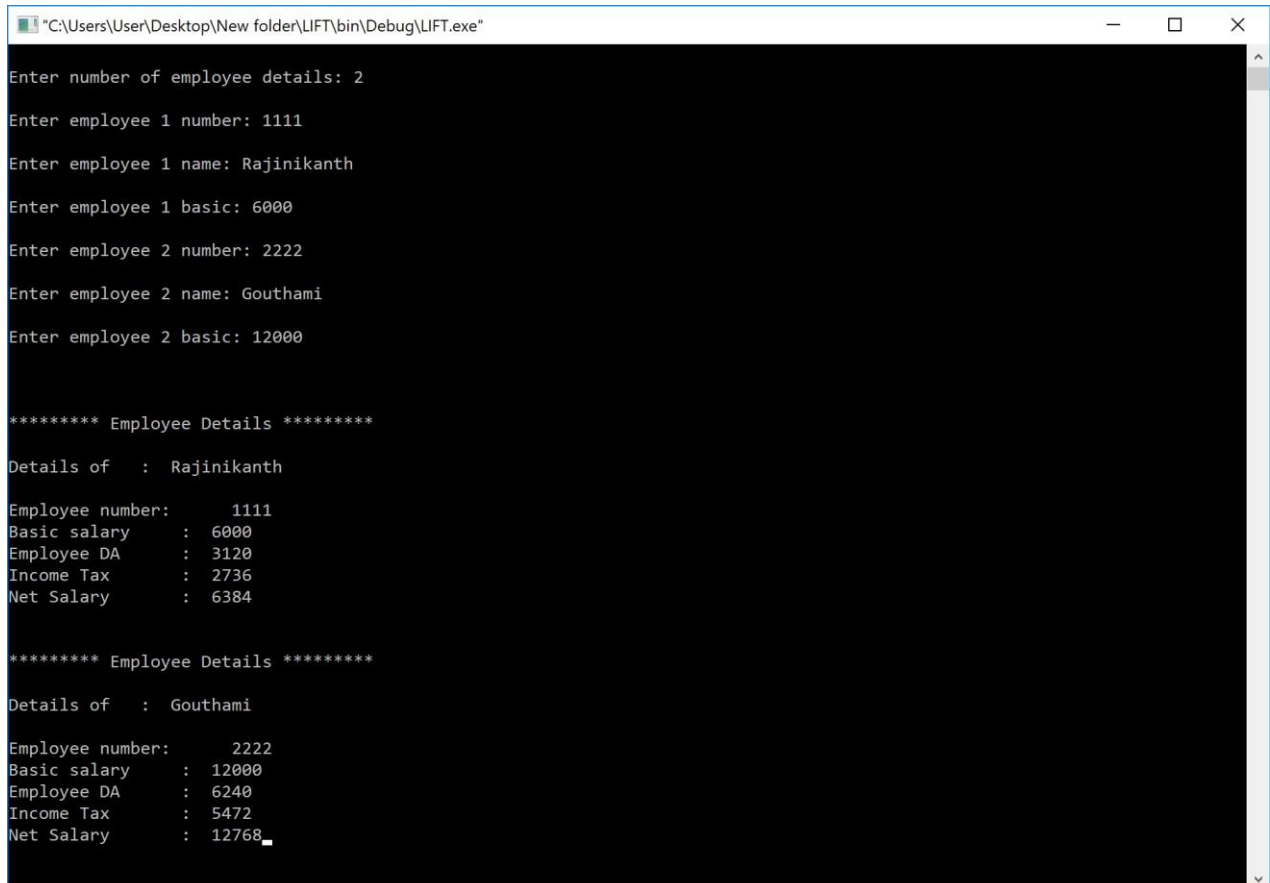
    for(i=0;i<num;i++)
        emp[i].find_net_sal();

    for(i=0;i<num;i++)
        emp[i].show_emp_details();

    getch();
    return 0;
}

```

Output:



```

C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe
Enter number of employee details: 2
Enter employee 1 number: 1111
Enter employee 1 name: Rajinikanth
Enter employee 1 basic: 6000
Enter employee 2 number: 2222
Enter employee 2 name: Gouthami
Enter employee 2 basic: 12000

***** Employee Details *****
Details of   : Rajinikanth
Employee number:    1111
Basic salary      : 6000
Employee DA       : 3120
Income Tax        : 2736
Net Salary        : 6384

***** Employee Details *****
Details of   : Gouthami
Employee number:    2222
Basic salary      : 12000
Employee DA       : 6240
Income Tax        : 5472
Net Salary        : 12768

```

Week6:

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

Source code:

```
#include <iostream>
#include<conio.h>

using namespace std;

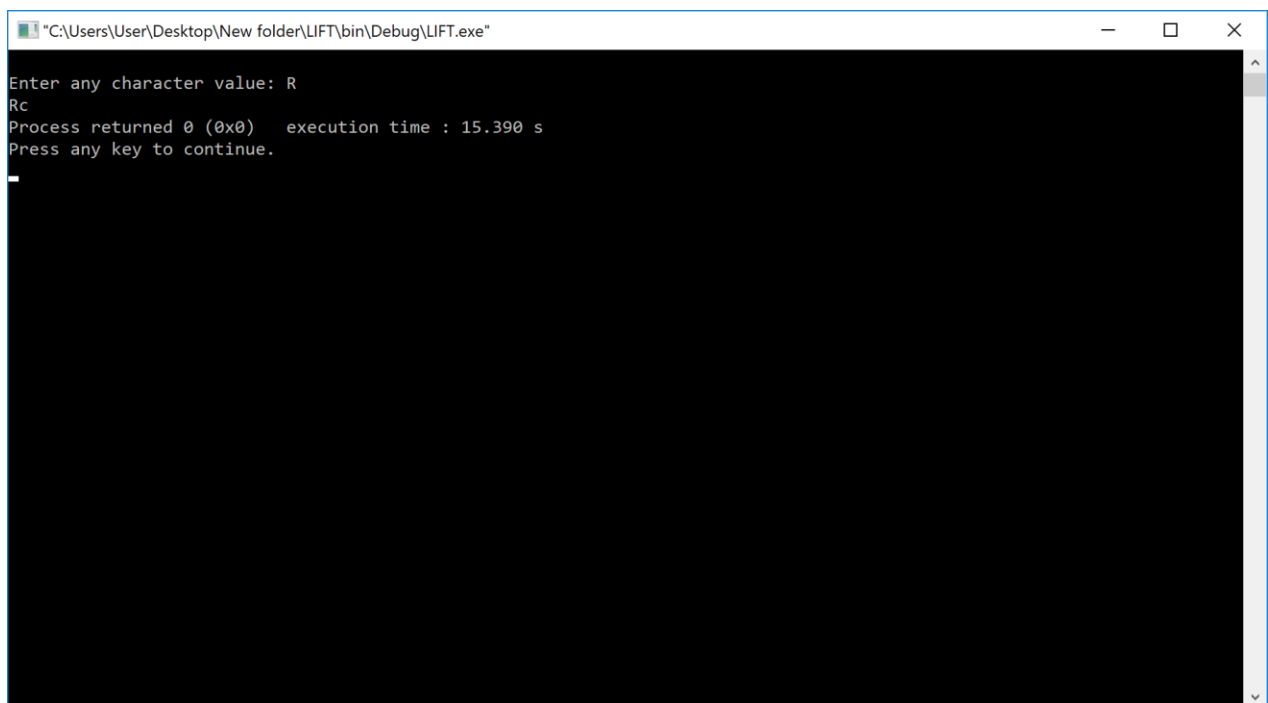
int main() {
    char c;

    cout<<"\nEnter any character value:";
    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;
}
```



```
"C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe"
Enter any character value: R
Rc
Process returned 0 (0x0) execution time : 15.390 s
Press any key to continue.
```

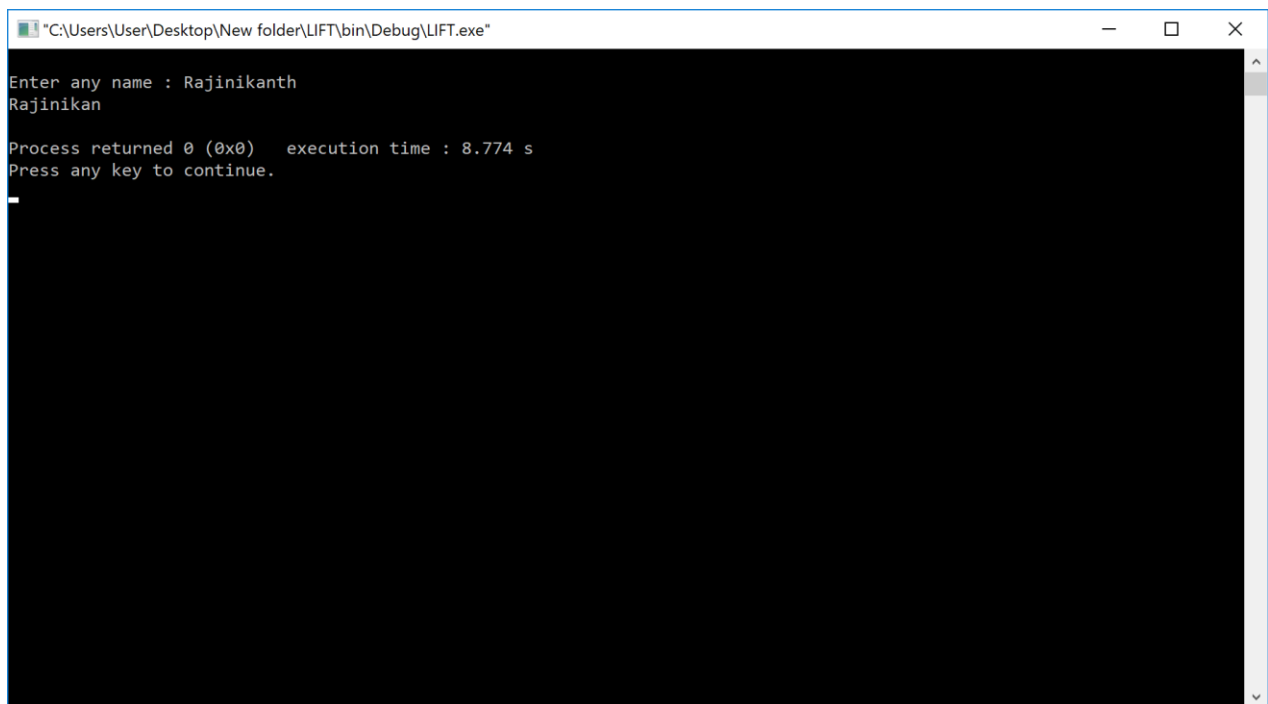
Output:

```
#include <iostream>
#include<conio.h>

using namespace std;

int main() {
    cout<<"\nEnter any name: ";
    char c[10];
    cin.getline(c,10); //It takes 10 characters as input;
    cout.write(c,9); //It reads only 9 character from buffer c;
    cout<<"\n";

    return 0;
}
```



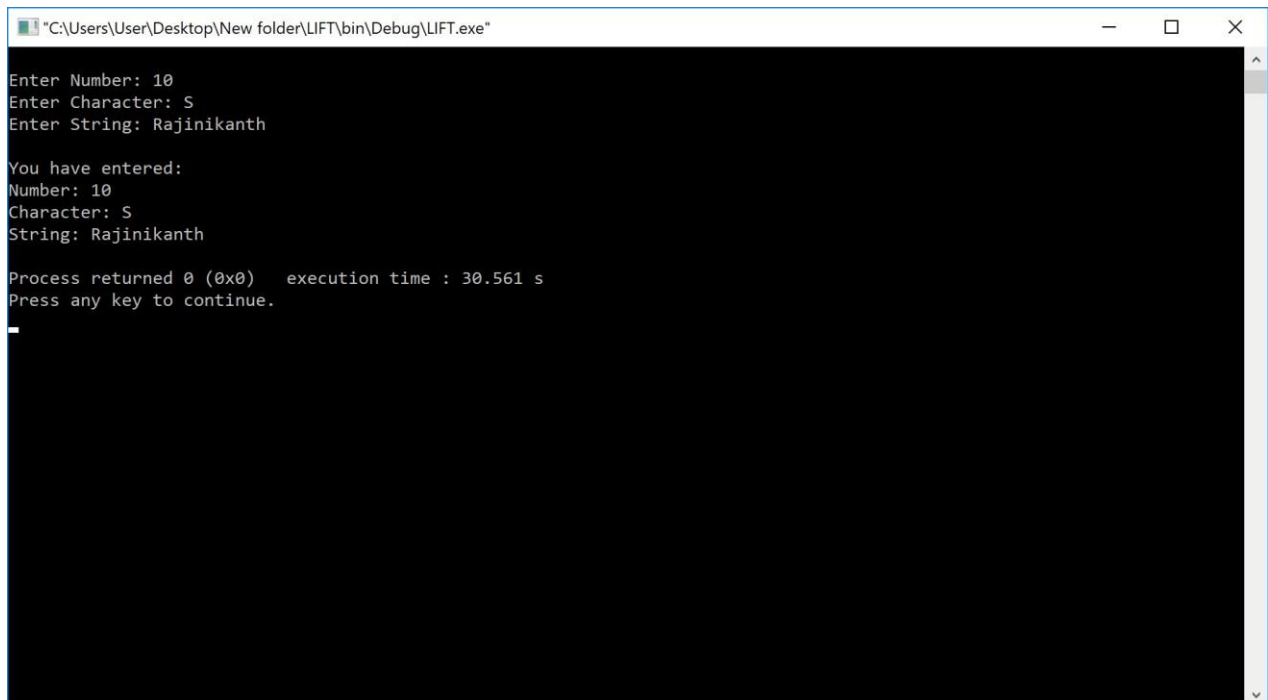
```
"C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe"
Enter any name : Rajinikanth
Rajinikan
Process returned 0 (0x0)   execution time : 8.774 s
Press any key to continue.
-
```

Output:

```
#include <iostream>
#include<conio.h>

using namespace std;

int main() {
    int num; char
    ch; stringstr;
    cout<<"\nEnterNumber:";
    cin>>num; //Inputs a variable;
    cout<<"Enter Character: ";
    cin>>ch; //Inputs a character;
    cout<<"Enter String: "; cin>>str;
    //Inputs astring;
    cout<<endl<<"Youhaveentered:\nNumber:"<<num<<"\nCharacter:"
        <<ch<<"\nString:"<<str<<endl;
    return 0;
}
```



```
"C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe"
Enter Number: 10
Enter Character: S
Enter String: Rajinikanth

You have entered:
Number: 10
Character: S
String: Rajinikanth

Process returned 0 (0x0) execution time : 30.561 s
Press any key to continue.
-
```

Output:

Week 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:

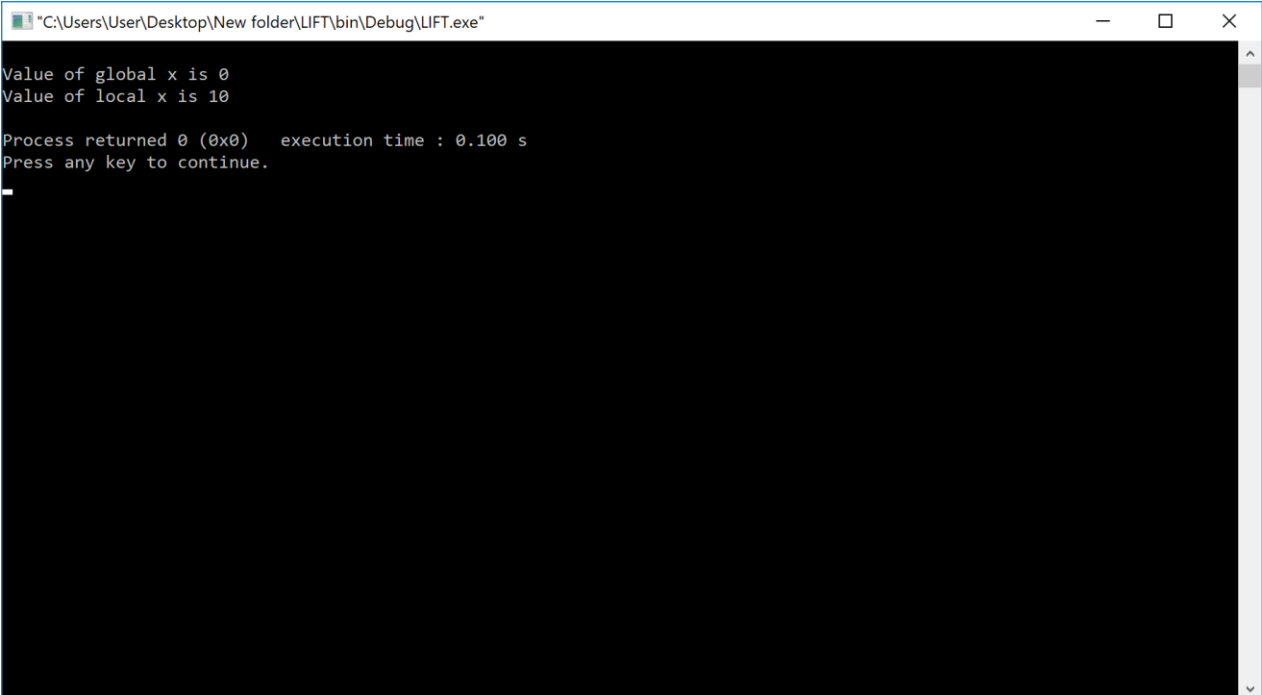
```
#include <iostream>
#include<conio.h>

using namespace std;

int x;

int main() {
    int x = 10; // Local x
    cout <<"\nValue of global x is " <<::x; cout
    <<"\nValue of local x is " << x<<endl;

    return 0;
}
```



```
"C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe"
Value of global x is 0
Value of local x is 10

Process returned 0 (0x0)   execution time : 0.100 s
Press any key to continue.
_
```

Output:

Week 8:

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

Source code:

```
#include <iostream>
#include<conio.h>

using namespace std;

int main() {

    double* pvalue = NULL; // Pointer initialized with null
    pvalue =newdouble; // Request memory for thevariable

    *pvalue=29494.99; // Store value at allocated address
    cout<<"\nValueofpvalue:"<<*pvalue<<endl;

    delete pvalue; // free up thememory.

    return 0;
}
```

```
"C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe"
Value of pvalue : 29495
Process returned 0 (0x0) execution time : 0.100 s
Press any key to continue.
```

Output:

Week 9:

Write a C++ program to create multilevel inheritance. (Hint: Classes A1, A2, A3)

Source code:

```
#include <iostream>
#include<conio.h>

using namespace std;

class base //single base class
{
    public:
    int x;
    void getdata() {
        cout <<"Enter value of x="; cin >>x;
    }
};
class derive1 : public base // derived class from base class
{
    public:
    int y;
    void readdata()
    {
        cout <<"\nEnter value of y= "; cin >>y;
    }
};
class derive2 :publicderive1 // derived from classderive1
{
    private:
    int z;
    public:
    void indata() {
        cout <<"\nEnter value of z= "; cin >>z;
    }
    void product() {
        cout<<"\nProduct="<<x*y*z<<endl;
    }
};

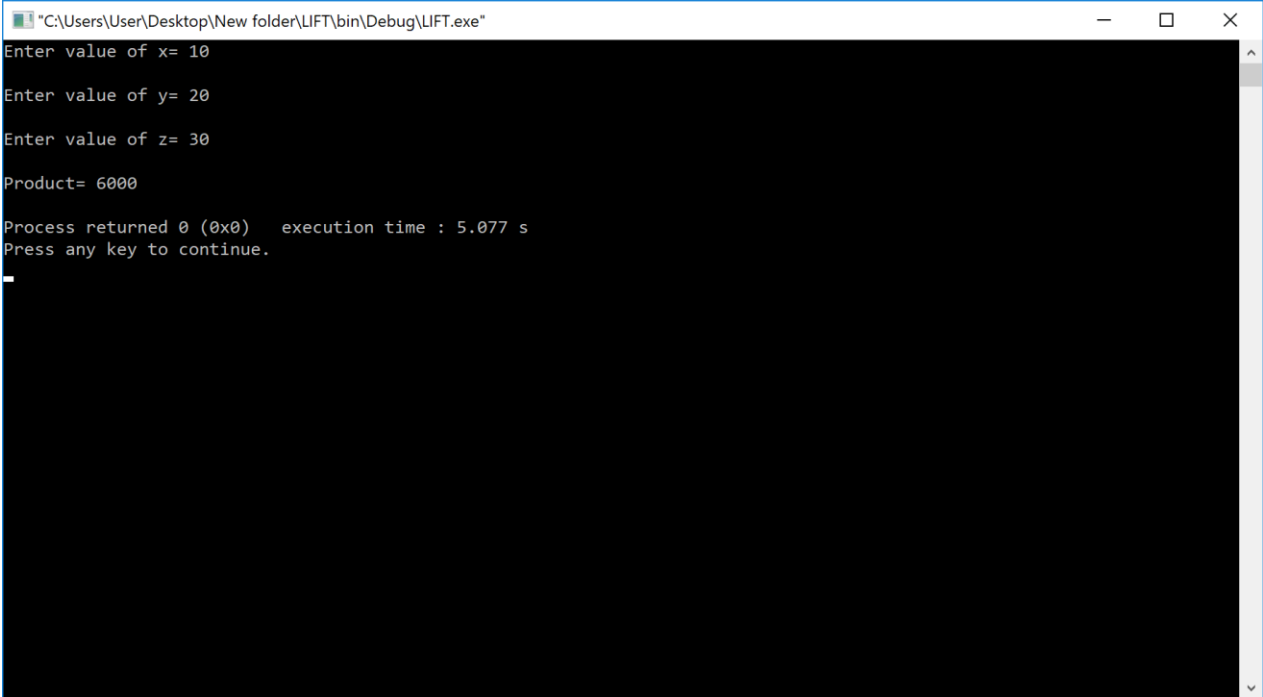
int main() {
```

```
derive2a;           //object of derived class
a.getdata();
a.readdata();
a.indata();
a.product();

return 0;

}
```

Output:



```
"C:\Users\User\Desktop\New folder\LIFT\bin\Debug\LIFT.exe"
Enter value of x= 10
Enter value of y= 20
Enter value of z= 30
Product= 6000
Process returned 0 (0x0)   execution time : 5.077 s
Press any key to continue.
-
```

Viva Questions:

1. What is Inheritance?
2. Types of inheritance?

Week 10:

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

Source code:

```
#include
<iostream>#include
<string>

using namespace std;

class Student
{
    string name; int
    marks; public:
    void getName()
    {
        getline( cin, name);
    }
    void getMarks()
    {
        cin >> marks;
    }
    void displayInfo()
    {
        cout <<"Name : " << name <<endl; cout
        <<"Marks : " << marks <<endl;
    }
};

int main()
{
    Student      st[5].*ptr;
    ptr=&st;
    for( int i=0; i<5; i++ )
    {
        cout <<"Student " <<i + 1 << endl; cout
        <<"Enter name" << endl;
        st[i]->getName();
        cout <<"Enter marks" << endl;
        st[i]->getMarks();
    }

    for( int i=0; i<5; i++ )
    {
        cout <<"Student " <<i + 1 <<endl;
        st[i]->displayInfo();
    }
    return 0;
}
```

Output:

Output

```
Student 1
Enter name
Jack
Enter marks
54
Student 2
Enter name
Marx
Enter marks
45
Student 3
Enter name
Julie
Enter marks
47
Student 4
Enter name
Peter
Enter marks
23
Student 5
Enter name
Donald Enter
marks 87
Student 1 Name
: Jack Marks :54
Student 2 Name
: Marx Marks
:45
Student 3 Name
: Julie Marks :47
Student 4 Name
: Peter Marks :23
Student 5 Name
: Donald Marks
:87
```

Week 11:

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

Source code:

```
#include <iostream> using
namespace std;

class Weapon{

    public:
        virtual void features() { cout <<"Loading weapon features.\n"; }
};

class Bomb : public Weapon {
    public:
        void features() {
            this->Weapon::features();
            cout <<"Loading bomb features.\n";
        }
};

class Gun : public Weapon {

    public:
        void features() {
            this->Weapon::features();
            cout <<"Loading gun features.\n";
        }
};

class Loader {

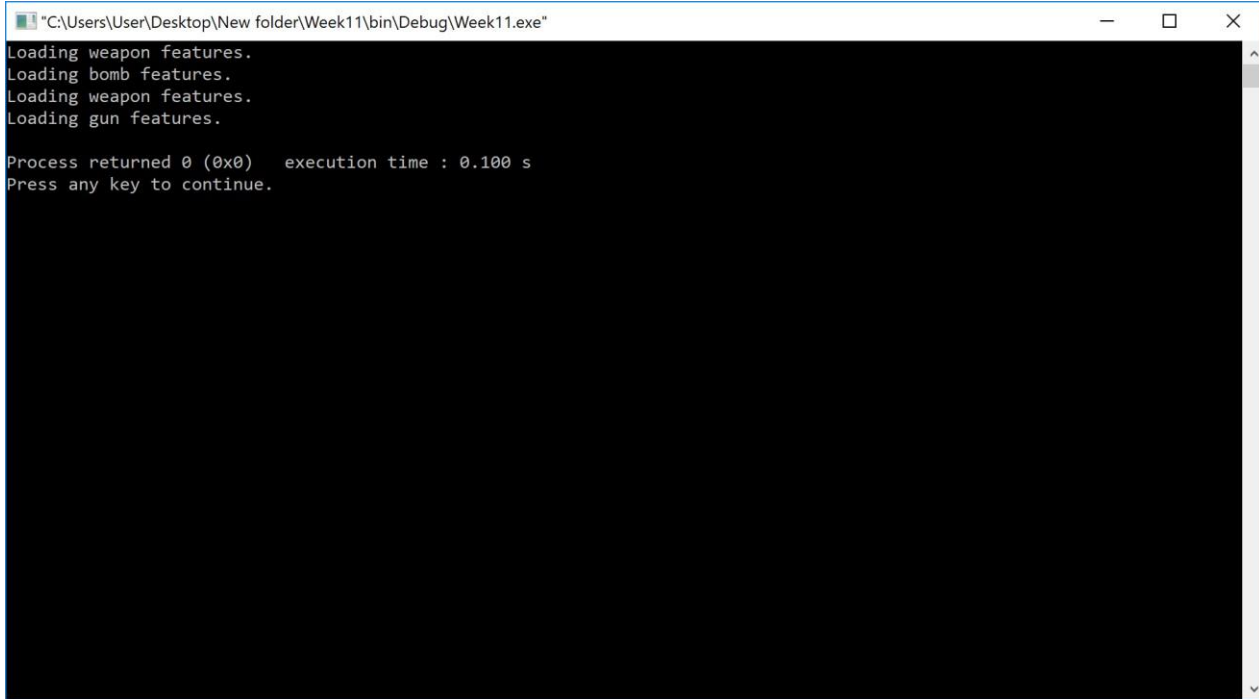
    public:
        void loadFeatures(Weapon *weapon) {
            weapon->features();
        }
};

int main() {
    Loader *l = new Loader;
    Weapon*w;
    Bomb b;
    Gun g; w
    =&b;
    l->loadFeatures(w);
    w =&g;
    l->loadFeatures(w);
}
```



```
    return 0;  
}
```

Output:



```
"C:\Users\User\Desktop\New folder\Week11\bin\Debug\Week11.exe"
Loading weapon features.
Loading bomb features.
Loading weapon features.
Loading gun features.

Process returned 0 (0x0)   execution time : 0.100 s
Press any key to continue.
```