2. OOP in C++, 3rd Edition, T. Gaddis, J. Walters and G. Muganda, Wiley Dream Tech Press.

3. Object Oriented Programming in C++, 3rd Edition, R. Lafore, Galigotia Publications Pvt Ltd.

## Content of Lab Experiments

| S.No. | Programs | Page No. |
|---|---|---|
| 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. | 10-11 |
| 2 | Write a C++ program to declare Struct. Initialize and display contents of member variables. | 12 |
| 3 | Write a C++ program to declare a class. Declare pointer to class. Initialize and display the contents of the class member. | 13 |
| 4 | Given that an EMPLOYEE class contains following members: data members: Employee number, Employee name, Basic, DA, IT, Net Salary and print data members. | 14-15 |
| 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). | 16-17 |
| 6 | Write a C++ to illustrate the concepts of console I/O operations. | 18 |
| 7 | Write a C++ program to use scope resolution operator. Display the various values of the same | 19 |
| 8 | Write a C++ program to allocate memory using new operator. | 20 |
| 9 | Write a C++ program to create multilevel inheritance. (Hint: Classes A1, A2, A3) | 21-22 |
| 10 | Write a C++ program to create an array of pointers. Invoke functions using array objects. | 23-24 |
| 11 | Write a C++ program to use pointer for both base and derived classes and call the member function. Use Virtual keyword. | 25 |

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.**

```
#include <iostream>
using namespace std;
#define MAX 10

class student
{
   private:
      char  name[30];
      int   rollNo;
      int   total;
      float perc;
   public:
      void getDetails(void);          //member function to get student's details
      void putDetails(void);       //member function to print student's details
};

void student:: getDetails(void)          //member function definition, outside of the class
{
   cout << "Enter name: " ;
   cin >> name;
   cout << "Enter roll number: ";
   cin >> rollNo;
   cout << "Enter total marks outof 500: ";
   cin >> total;
   perc=(float)total/500*100;
}

void student:: putDetails(void)          //member function definition, outside of the class
{
   cout << "Student details:\n";
   cout << "Name:"<< name << ",Roll Number:" << rollNo << ",Total:" << total << ",Percentage:" <<
              perc;
}

int main()
{
    student std[MAX];       //array of objects creation
    int n,loop;
    cout << "Enter total number of students: ";
    cin >> n;
    for (loop=0;loop< n; loop++)
     {
        cout << "Enter details of student " << loop+1 << ":\n";
        std[loop].getDetails();
     }
    cout << endl;
    for(loop=0;loop< n; loop++)
     {
        cout << "Details of student " << (loop+1) << ":\n";
        std[loop].putDetails();
     }
   return 0;
}
```

**Output**

```
    Enter total number of students: 3
    Enter details of student 1:
    Enter name: Karthik
    Enter roll number: 1201
    Enter total marks out of 500: 456

    Enter details of student 2:
    Enter name: Mahesh
    Enter roll number: 1202
    Enter total marks out of 500: 398

    Enter details of student 3:
    Enter name: Kiran
    Enter roll number: 1203
    Enter total marks out of 500: 456

    Details of student 1:
    Student details:
    Name: Karthik, Roll Number: 101, Total: 456, Percentage: 91.2

    Details of student 2:
```

```
Student details:
Name: Mahesh, Roll Number: 1202, Total: 398, Percentage:79.6

Details of student 3:
Student details:
Name: Kiran, Roll Number: 1203, Total: 398, Percentage:79.6
```

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

```
#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 member.**

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

class Box
{
  public:
          Box(double l = 2.0, double b = 2.0, double h = 2.0)
          {
          cout <<"Constructor called." << endl;
          length = l;
          breadth = b;
          height = h;
          }
          double Volume()
          {
          return length * breadth * height;
          }
  private:
          double length;
          double breadth;
          double height;
};

int main(void)
{
  Box Box1(3.3, 1.2, 1.5);
  Box Box2(8.5, 6.0, 2.0);
  Box *ptrBox;
  ptrBox = &Box1;
  cout << "Volume of Box1: " << ptrBox->Volume() << endl;
  ptrBox = &Box2;
  cout << "Volume of Box2: " << ptrBox->Volume() << endl;
  return 0;
}
```

**Output:**

```
Constructor called.
Constructor called.
Volume of Box1: 5.94
Volume of Box2: 102
```

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

```
#include<iostream.h>
#include<conio.h>
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();
                void find_net_sal();
                void show_emp_details();
};
void employee :: get_details()
{
        cout<<"\n Enter employee number:\n";
        cin>>emp_num;
        cout<<"\n Enter employee name:\n";
        cin>>emp_name;
        cout<<"\n Enter employee basic:\n";
        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 Details of   :  "<<emp_name;
        cout<<"\n\n Employee number:      "<<emp_num;
        cout<<"\n Basic salary     :  "<<emp_basic;
        cout<<"\n Employee DA      :  "<<emp_da;
        cout<<"\n Income Tax       :  "<<emp_it;
        cout<<"\n Net Salary       :  "<<net_sal;
}
int main()
{
        employee emp[10];
        int i,num;
        clrscr();
        cout<<"\n Enter number of employee details\n";
        cin>>num;
        for(i=0;i<num;i++)
                emp[i].get_details();
        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:**

```
Enter number of employee details
Enter employee number: 5123
Enter employee name: Madhav
Enter employee basic: 10000

Details of   :  Madhav
Employee number:  5123
Basic salary     :    10000
Employee DA      :  5200
Income Tax       :  4560
Net Salary       :  10640
```

**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).**

```
#include<iostream.h>
#include<conio.h>
#define SIZE 5
class emp
 {
    float basic,da,it,netsal;
    char name[20],num[10];
    public:
       void getdata();
       void net_sal();
       void dispdata();
 };
void emp::getdata()
   {
    cout<<"\n  Enter employee number: " ;
    cin>>name;
    cout<<"\n  Enter employee name: " ;
    cin>>num;
    cout<<"Enter employee basic salary in Rs: " ;
    cin>>basic;
   }
void emp::net_sal()
   {
    da=((0.52)*basic );
    float gsal=da+basic;
    it=((0.3)*gsal);
    netsal=gsal-it;
   }

void emp::dispdata()
   {
    cout <<"\n  Employee number: "<<name
    cout <<"\n  Employee name: "<<num
    cout <<"\n  Employee netsalary: "<<netsal<<" Rs.";
   }
void main()
 {
    clrscr();
    emp ob[SIZE];
    int n;
    cout<<"\n\n***********************************"
      <<"\n Calculation of Employee Net Salary"
      <<"\n***********************************"
      <<"\n Enter the number of employees";
    cin>>n;
    for(int i=0;i<n;i++)
      {
        ob[i].getdata();
        ob[i].net_sal();
      }
    clrscr();
    cout<<"\n-----------------"
      <<"\n Employee Detail::"
      <<"\n-----------------";
    for( i=0;i<n;i++)
       {
        cout<<"\n\n Employee:"<<i+1
           <<"\n ----------";
        ob[i].dispdata();
       }
    getch();
  }
```

**Output:**

```
***********************************"
Calculation of Employee Net Salary
***********************************"
Enter the number of employees: 1

Enter employee number: 22
Enter employee name: Sanath
Enter employee basic salary in Rs: 10000

-----------------
Employee Detail::
-----------------
Employee:1
Employee number: 22
Employee name: Sanath
Employee netsalary: 10000 RS.
```

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

```
#include <iostream>
#include <fstream>
#include <cstdlib>
#include <string>
using namespace std;

int main()
{
  string filename = "test.txt";
  ofstream fout(filename.c_str());  // default mode is ios::out | ios::trunc

  if (!fout)
  {
    cerr << "error: open file for output failed!" << endl;
    abort();  // in <cstdlib> header
  }

  fout << "apple" << endl;
  fout << "orange" << endl;
  fout << "banana" << endl;
  fout.close();
  ifstream fin(filename.c_str());  // default mode ios::in

  if (!fin)
  {
    cerr << "error: open file for input failed!" << endl;
    abort();
  }
  char ch;

  while (fin.get(ch))
   {  // till end-of-file
    cout << ch;
  }
  fin.close();
  return 0;
}
```

**Output:**

```
apple
orange
banana
```

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

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

class programming
{
    public: void output();  //function declaration
};
void programming::output()
{
 cout << "Function defined outside the class.\n";
}

int main()
{
    programming x;
    x.output();
    return 0;
 }
```

**Ouput:**

```
Function defined outside class
```

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

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

int main ()
{
  int* p = NULL;
  p = new(nothrow) int;
   if (!p)
     cout << "allocation of memory failed\n";
   else
   {
     *p = 29;
     cout << "Value of p: " << *p << endl;
   }
  float *r = new float(75.25);
  cout << "Value of r: " << *r << endl;
  int n = 5;
  int *q = new(nothrow) int[n];

  if (!q)
     cout << "allocation of memory failed\n";
  else
  {
    for (int i = 0; i < n; i++)
        q[i] = i+1;
     cout << "Value store in block of memory: ";
     for (int i = 0; i < n; i++)
        cout << q[i] << " ";
   }
  delete p;
  delete r;
  delete[] q;
```

```
   return 0;
}
```

**Output:**

```
Value of p: 29

Value of r: 75.25

Value store in block of memory: 1 2 3 4 5
```

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

```
#include <iostream>
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 : public derive1   // derived from class derive1
{
        private:
        int z;
        public:
        void indata()
        {
        cout << "\nEnter value of z= "; cin >> z;
```

```
            }
            void product()
            {
               cout << "\nProduct= " << x * y * z;
            }
};




int main()
{
    derive2 a;
    a.getdata();
    a.readdata();
    a.indata();
    a.product();
    return 0;
}
```

**Output:**

```
Enter value of x= 2
Enter value of y= 3
Enter value of z= 3
Product= 18
```

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

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

class A
{
public:
virtual void show()
 {
  cout<<“A\n”;
 }
};

class B : public A
{
public:
 void show()
  {
    cout<<“B\n”;
   }
 };

class C : public A
{
  public:
   void show()
    {
     cout<<“C\n”;
    }
};
```

```
class D : public A
{
  public:
   void show()
   {
   cout<<“D\n”;
   }
};




class E : public A
{
 public:
  void show()
  {
   cout<<“E”;
  }
};

void main()
{
clrscr();
A a;
B b;
C c;
D d;
E e;
A *p a[]={&a,&b,&c,&d,&e};
for ( int j=0;j<5;j++)
pa[j]->show();
}
```

**Output:**

```
A
B
C
D
E
```

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

```
#include<iostream>
using namespace std;
class base
{
public:
    virtual void print ()
    {
       cout<< "print base class" <<endl;
     }
  void show ()
    {
        cout<< "show base class" <<endl;
     }
};
class derived: public base
{
public:
    void print ()
    {
       cout<< "print derived class" <<endl;
     }

    void show ()
    {
       cout<< "show derived class" <<endl;
     }
};

int main()
{
    base *bptr;
    derived d;
    bptr = &d;
   bptr->print();
   bptr->show();
}
```

**Output:**

```
print derived class
show base class
```