GOA COLLEGE OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

SUBJECT: - OOPS

FACULTY: - Prof. AMIT P. PATIL & Prof. NITESH NAIK **CLASS: - SE Comp (III)**

PLATFORM: - Dev C++/VS 2010 **YEAR: - 31-7-23 to DEC 23**

Experiment 1

Aim: To study basics of C++ programming

Theory:

1. Classes and Objects in C++:

In C++, classes are blueprints for creating objects. A class defines the structure and behavior of an object, including its attributes (data members) and methods (functions). Objects are instances of a class, representing real-world entities. They encapsulate data and functions, promoting code organization and reusability.

2. Array of Objects:

In C++, you can create arrays of objects just like arrays of primitive data types. An array of objects allows you to store multiple objects of the same class in a contiguous memory block. This is useful for managing collections of similar entities, simplifying operations like iteration and manipulation.

3. Static:

In C++, "static" has multiple meanings. When applied to a variable within a function, it makes that variable retain its value across function calls. When applied to a class member (data or function), it's shared among all instances of that class, and it can be accessed using the class name without creating an object.

4. Dynamic Memory Allocation:

Dynamic memory allocation in C++ refers to the process of allocating memory for objects at runtime, using operators like `new` (to allocate) and `delete` (to deallocate). This allows you to create objects on the heap, providing more flexibility compared to objects on the stack (automatic duration). Proper management of dynamic memory is essential to prevent memory leaks and optimize resource usage.

Questions:

A] Write a C++ program to implement a student management system. Record the following details of the student in your program: Name, roll_no, branch,marks of 5 subjects(maths,ld,ds,oops,co).roll_no to be allocated by the program sequentially(auto).Create a menu driven program(switch case) to prompt the user for following options.
Option 1: To enter 3 students details(create static array of object for 3 students).

Option 2: display a student details based on roll_no entered.
Option 3: Compute result(%) of a specific student based on roll_no entered and display it on screen.

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

#define SPACE (cout<<endl<<endl)

struct marks {
   double maths;
   double ld;
   double co;
   double ds;
   double oops;
   double percentage;
};

class student
{
   string name;
   int rollno;
   string branch;
   struct marks m;
public:
   void EnterDetails(int base);
   bool DisplayDetails(int roll);
   bool DisplayPer(int roll);
};

int main()
{
   bool loop = true;
   int op=0;
   int base=2200;
   int roll;
   bool flag;

   while(loop){
      cout<<"*****MENU*****"<<endl;
      cout<<"1.Enter 3 student detail\n2.Diplay Stuent details\n3.Display Percentage\n4.Exit"<<endl;
      cin>>op;
      switch(op)
      {
         case 1:
            static student s[3];
               for(int i=0; i<3; i++){
                  base++;
                  s[i].EnterDetails(base);
               }
            break;
         case 2:
            cout<<"Enter Roll No.: "; cin>>roll;
            for(int i=0; i<3; i++){
               if(s[i].DisplayDetails(roll)==true){
                  flag = true;
```

```
                break;
            }
        }
        if(flag!=true) cout<<"Roll No. not found"<<endl;
        flag = true;
        break;
      case 3:
        cout<<"Enter Roll No.: "; cin>>roll;
        for(int i=0; i<3; i++){
          if(s[i].DisplayPer(roll)==true){
            flag = true;
            break;
          }
        }
        if(flag!=true) cout<<"Roll No. not found"<<endl;
        flag = true;
        break;
      case 4: loop = false; break;

      default : cout<<"ERROR! Wrong input!"<<endl;
    }
  }
}

void student :: EnterDetails(int base)
{
  cout<<"Enter Student name: ";cin>>name;
  cout<<"Enter Branch: "; cin>>branch;
  cout<<"Enter markss of maths, LD, CO, DS, OOPS: ";
  cin>>m.maths>>m.ld>>m.co>>m.ds>>m.oops;
  rollno = base;
  cout<<"Roll No. : "<<rollno<<endl;
  DisplayDetails(rollno);
  SPACE;
}
bool student :: DisplayDetails(int roll)
{
  if(roll != rollno) return false;
  cout<<"Student name: "<<name<<endl; cout<<"Branch: "<<branch<<endl;
  cout<<"Maths : "<<m.maths<<endl; cout<<"LD : "<<m.ld<<endl;
  cout<<"CO : "<<m.co<<endl; cout<<"DS : "<<m.ds<<endl; cout<<"OOPS : "<<m.oops<<endl;
  return true;
  SPACE;
}
bool student :: DisplayPer(int roll)
{
  if(roll != rollno) return false;
  m.percentage = (m.maths + m.ld + m.co + m.ds + m.oops)/5;
  cout<<"Percentage : "<<m.percentage<<endl;
  return true;
  SPACE;
}
```

Output :

```
*****MENU*****
1.Enter 3 student detail
```

2.Diplay Stuent details
3.Display Percentage
4.Exit
1
Enter Student name: divyam
Enter Branch: computer
Enter markss of maths, LD, CO, DS, OOPS: 80
90
70
60
70
Roll No. : 2201
Student name: divyam
Branch: computer
Maths : 80
LD : 90
CO : 70
DS : 60
OOPS : 70

Enter Student name: kevin
Enter Branch: computer
Enter markss of maths, LD, CO, DS, OOPS: 20
45
67
89
34
Roll No. : 2202
Student name: kevin
Branch: computer
Maths : 20
LD : 45
CO : 67
DS : 89
OOPS : 34

Enter Student name: eren
Enter Branch: computer
Enter markss of maths, LD, CO, DS, OOPS: 45
78
90
5
100
Roll No. : 2203
Student name: eren
Branch: computer
Maths : 45
LD : 78
CO : 90
DS : 5
OOPS : 100

*****MENU*****
1.Enter 3 student detail

```
2.Diplay Stuent details
3.Display Percentage
4.Exit
2
Enter Roll No.: 2201
Student name: divyam
Branch: computer
Maths : 80
LD : 90
CO : 70
DS : 60
OOPS : 70
*****MENU*****
1.Enter 3 student detail
2.Diplay Stuent details
3.Display Percentage
4.Exit
3
Enter Roll No.: 2201
Percentage : 74
*****MENU*****
1.Enter 3 student detail
2.Diplay Stuent details
3.Display Percentage
4.Exit
4
```

B] Same program implementation using DYNAMIC MEMORY ALLOCATION (new operator)

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

#define SPACE (cout<<endl<<endl)

struct marks {
   double maths;
   double ld;
   double co;
   double ds;
   double oops;
   double percentage;
};

class student
{
   string name;
   int rollno;
   string branch;
   struct marks m;
public:
   void EnterDetails(int base);
   bool DisplayDetails(int roll);
   bool DisplayPer(int roll);
};
```

```
int main()
{
   bool loop = true;
   int op=0;
   int base=2200;
   int roll;
   bool flag;
   student* s = new student [3];

   while(loop){
      cout<<"*****MENU*****"<<endl;
      cout<<"1.Enter 3 student detail\n2.Diplay Stuent details\n3.Display Percentage\n4.Exit"<<endl;
      cin>>op;
      switch(op)
      {
         case 1:
               for(int i=0; i<3; i++){
                  base++;
                  s[i].EnterDetails(base);
               }
            break;
         case 2:
            cout<<"Enter Roll No.: "; cin>>roll;
            for(int i=0; i<3; i++){
               if(s[i].DisplayDetails(roll)==true){
                  flag = true;
                  break;
               }
            }
            if(flag!=true) cout<<"Roll No. not found"<<endl;
            flag = true;
            break;
         case 3:
            cout<<"Enter Roll No.: "; cin>>roll;
            for(int i=0; i<3; i++){
               if(s[i].DisplayPer(roll)==true){
                  flag = true;
                  break;
               }
            }
            if(flag!=true) cout<<"Roll No. not found"<<endl;
            flag = true;
            break;
         case 4: loop = false; break;

         default : cout<<"ERROR! Wrong input!"<<endl;
      }
   }
}

void student :: EnterDetails(int base)
{
   cout<<"Enter Student name: ";cin>>name;
   cout<<"Enter Branch: "; cin>>branch;
   cout<<"Enter markss of maths, LD, CO, DS, OOPS: ";
   cin>>m.maths>>m.ld>>m.co>>m.ds>>m.oops;
   rollno = base;
```

```
   cout<<"Roll No. : "<<rollno<<endl;
   DisplayDetails(rollno);
   SPACE;
}
bool student :: DisplayDetails(int roll)
{
   if(roll != rollno) return false;
   cout<<"Student name: "<<name<<endl; cout<<"Branch: "<<branch<<endl;
   cout<<"Maths : "<<m.maths<<endl; cout<<"LD : "<<m.ld<<endl;
   cout<<"CO : "<<m.co<<endl; cout<<"DS : "<<m.ds<<endl; cout<<"OOPS : "<<m.oops<<endl;
   return true;
   SPACE;
}
bool student :: DisplayPer(int roll)
{
   if(roll != rollno) return false;
   m.percentage = (m.maths + m.ld + m.co + m.ds + m.oops)/5;
   cout<<"Percentage : "<<m.percentage<<endl;
   return true;
   SPACE;
}
```

Output:

```
*****MENU*****
1.Enter 3 student detail
2.Diplay Stuent details
3.Display Percentage
4.Exit
1
Enter Student name: divyam
Enter Branch: computer
Enter markss of maths, LD, CO, DS, OOPS: 80
90
70
60
70
Roll No. : 2201
Student name: divyam
Branch: computer
Maths : 80
LD : 90
CO : 70
DS : 60
OOPS : 70


Enter Student name: kevin
Enter Branch: computer
Enter markss of maths, LD, CO, DS, OOPS: 20
45
67
89
34
Roll No. : 2202
Student name: kevin
Branch: computer
```

```
Maths : 20
LD : 45
CO : 67
DS : 89
OOPS : 34


Enter Student name: eren
Enter Branch: computer
Enter markss of maths, LD, CO, DS, OOPS: 45
78
90
5
100
Roll No. : 2203
Student name: eren
Branch: computer
Maths : 45
LD : 78
CO : 90
DS : 5
OOPS : 100


*****MENU*****
1.Enter 3 student detail
2.Diplay Stuent details
3.Display Percentage
4.Exit
2
Enter Roll No.: 2201
Student name: divyam
Branch: computer
Maths : 80
LD : 90
CO : 70
DS : 60
OOPS : 70
*****MENU*****
1.Enter 3 student detail
2.Diplay Stuent details
3.Display Percentage
4.Exit
3
Enter Roll No.: 2201
Percentage : 74
*****MENU*****
1.Enter 3 student detail
2.Diplay Stuent details
3.Display Percentage
4.Exit
4
```

Conclusion: Basic of C++ programming were studied. Two programs were written in C++.