EXPERIMENT NUMBER: 2.1

STUDENT'S NAME:

UID:

Section:

SEMESTER: 2nd

1. AIM: LEARN HOW TO USE CLASSES USING C++

PROGRAMS:

:Write a program that takes information about institute staff information for

1) Teacher code, name, subject and publication

2) Officer code, name and grade

3) Typist code, name, speed and daily wages and displays it using hierarchal inheritance.

: Create a class student having student uid and getnumber(),putnumber() as member functions to get the values and display it. Derive a class test having marks in different subjects and getmarks() and putmarks() as member functions to get and display the values. Derive another class sports from student class having sports score and getscore(), putscore() as member functions to get and display the values. Derive a class result from test and sports class and define a function display() to calculate total marks. Implement it with the object of result class. If it gives any error, resolve it by adding the required functionality.

: WAP to illustrate how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named alpha, beta, gamma such that alpha, beta are base class and gamma is derived class inheriting alpha &beta. Pass four variable to gamma class object which will further send one integer variable to alpha(),one float type variable to beta().Show the order of execution by invoking constructor of derived class.

2. PROGRAM CODE:

PROGRAM 5.1:

```
#include <iostream>

#include <conio.h>

using namespace std;

class staff

{

 protected:

     int code;

     char name[20];

 public:

     void getstaff(void)

     {

   cout<<"\n\nEnter code :-";

   cin>>code;

   cout<<"Enter name :-";

   cin>>name;

     }

     void dispstaff(void)

     {

      cout<<"\nNAME       :-

"<<name;
```

```
     cout<<"\nCODE      :-
"<<code;
    }
};
class teacher : public staff
{
    char sub[20];
    char pub[20];
 public:
    void create(void)
    {
  getstaff();
  cout<<"Enter Subject :-";
  cin>>sub;
  cout<<"\nEnter Publication :-
";
  cin>>pub;
    }
    void display(void)
    {
   dispstaff();
   cout<<"\nSUBJECT   :-
"<<sub;
```

```
    cout<<"\nPUBLICATION:-

"<<pub;

     }

};

class officer : public staff

{

      char grade;

  public:

   void create(void)

   {

      getstaff();

      cout<<"Enter Grade :-";

      cin>>grade;

   }

   void display(void)

   {

      dispstaff();

      cout<<"\nGRADE      :-

"<<grade;

   }

};

class typist : public staff

{
      float speed;
 public:
     void gettypist(void)
     {
   getstaff();
   cout<<"Enter speed (wpm):-";
   cin>>speed;
     }
     void disptypist(void)
     {
   dispstaff();
   cout<<"\nSPEED     :-
"<<speed;
     }
};
class casual : public typist
{
    float dailywages;
  public:
      void create(void)
   {
     gettypist();
     cout<<"Enter Daily Wages :-
```

```
";
     cin>>dailywages;
   }
   void display(void)
   {
     disptypist();
     cout<<"\nDAILY WAGES:-
"<<dailywages;
   }
};
int main()
{
teacher o1t[10];
casual o1c[10];
officer o1o[10];
int choice,i;
char test;
while(1)
{
int count;
start:

cout<<"\n=====EDUCATION
```

```
INSTITUTION

DATABASE=====\n\n\n";

  cout<<"Choose Category of

Information\n";

  cout<<"1)  Teachers\n";

  cout<<"2)  Officer\n";

  cout<<"3)  Typist\n";

  cout<<"4)  Exit\n";

  cout<<"Enter your choice:-";

  cin>>choice;

  switch(choice)

  {

     case 1 : while(1)

       {


cout<<"\n=====TEACHERS

INFORMATION=====\n\n";

       cout<<"\nChoose your

choice\n";

       cout<<"1) Create\n";

       cout<<"2) Display\n";

       cout<<"3) Jump to Main

Menu\n";
```

```
       cout<<"Enter your choice:-

";

       cin>>choice;

       switch(choice)

       {

        case 1 :

for(count=0,i=0;i<10;i++)

               {

                cout<<endl;

                o1t[i].create();

                count++;

                cout<<endl;

                cout<<"\n\nAre you

Interested in entering data\n";

                cout<<"Enter y or n:-

";

                cin>>test;

                if(test=='y' ||

test=='Y')

               continue;

                else goto out1;

                }

                out1:
```

```
              break;

         case 2 :

    for(i=0;i<count;i++)

              {

               cout<<endl;

                o1t[i].display();

                cout<<endl;

              }

              getch();

              break;

         case 3 : goto start;

         default: cout<<"\nEnter

    choice is invalid\ntry

    again\n\n";

         }

         }

      case 2 :  while(1)

         {


    cout<<"\n=====OFFICERS

    INFORMATION=====\n\n";

        cout<<"\nChoose your

    choice\n";
```

```
       cout<<"1) Create\n";

       cout<<"2) Display\n";

       cout<<"3) Jump to Main

Menu\n";

       cout<<"Enter your choice:-

";

       cin>>choice;

       switch(choice)

       {

        case 1 :

for(count=0,i=0;i<10;i++)

               {

                cout<<endl;

                o1o[i].create();

                 count++;

                 cout<<endl;

                 cout<<"\n\nAre you

Interested in entering data\n";

                 cout<<"Enter y or n:-

";

                 cin>>test;

                 if(test=='y' ||

test=='Y')
```

```
               continue;

                else goto out2;

                }

                out2:

                break;

        case 2 :

for(i=0;i<count;i++)

              {

               cout<<endl;

                o1o[i].display();

                cout<<endl;

              }

              getch();

              break;

        case 3 : goto start;

        default: cout<<"\nInvalid

choice\ntry again\n\n";

        }

        }

     case 3 : while(1)

       {

       cout<<"\n=====TYPIST

INFORMATION=====\n\n";
```

```
       cout<<"\nChoose your
choice\n";

       cout<<"1) Create\n";

       cout<<"2) Display\n";

       cout<<"3) Jump to Main
Menu\n";

       cout<<"Enter your choice:-
";

       cin>>choice;

       switch(choice)

       {

        case 1 :
for(count=0,i=0;i<10;i++)

              {

                cout<<endl;

                o1c[i].create();

                count++;

                cout<<endl;

                cout<<"\n\nAre you
Interested in entering data\n";

                cout<<"Enter y or n:-
";

                cin>>test;
```

```
      if(test=='y' ||
test=='Y')
     continue;
      else goto out3;
      }
      out3:
      break;
   case 2 :
for(i=0;i<count;i++)
     {
      cout<<endl;
      o1c[i].display();
      cout<<endl;
     }
     getch();
     break;
   case 3 : goto start;
   default: cout<<"\nInvalid
choice\ntry again\n\n";
   }
   }
  case 4 : goto end;
 }
```

```
}

end:

        return 0;

}
```

PROGRAM 5.2:

```
   #include<iostream>
#include<conio.h>
using namespace std;
class student {
  int rno;
public:
  void getnumber() {
     cout << "Enter Roll No:";
     cin>>rno;
  }
  void putnumber() {
     cout << "\n\n\tRoll No:" << rno << "\n";
  }
};
class test : virtual public student {
public:
  int part1, part2;
  void getmarks() {
     cout << "Enter Marks\n";
     cout << "Part1:";
     cin>>part1;
     cout << "Part2:";
     cin>>part2;
  }
  void putmarks() {
     cout << "\tMarks Obtained\n";
     cout << "\n\tPart1:" << part1;
     cout << "\n\tPart2:" << part2;
  }
};
class sports : public virtual student {
public:
  int score;
  void getscore() {
     cout << "Enter Sports Score:";
     cin>>score;
  }
  void putscore() {
     cout << "\n\tSports Score is:" << score;
  }
};
class result : public test, public sports {
  int total;
public:
  void display() {
     total = part1 + part2 + score;
     putnumber();
     putmarks();
     putscore();
     cout << "\n\tTotal Score:" << total;
  }
};
int main() {
  result obj;
  obj.getnumber();
  obj.getmarks();
  obj.getscore();
  obj.display();
  return 0;
}
```

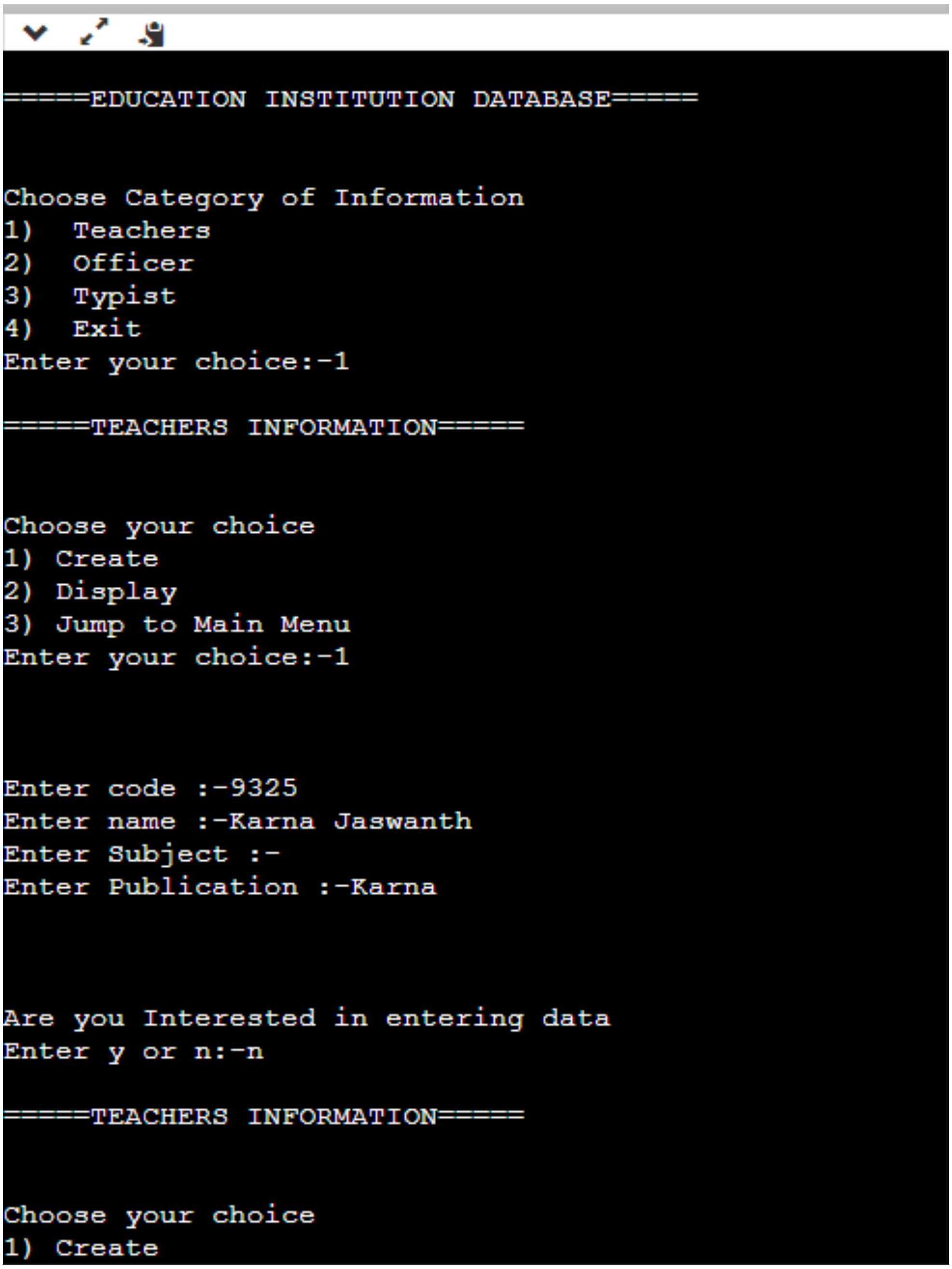
PROGRAM 5.3:

```
#include<iostream>
#include<conio.h>
using namespace std;
class alpha
{
        int x;
        public:
                alpha(int i)
                {
                        x=i;
                        cout<<"alpha initialized\n";
                }
                void show_x()
                {
                        cout<<"x="<<x<<"\n";
                }
};
class beta
{
        float y;
        public:
```

```
        beta(float j)
        {
            y=j;
            cout<<"beta initialized\n";
        }
        void show_y()
        {
            cout<<"y="<<y<<"\n";
        }
};
class gamma : public beta,public alpha
{
    int m,n;
    public:
        gamma(int a,float b,int c,int d): alpha(a),beta(b)
        {
            m=c,n=d;
            cout<<"gamma initialized\n";
        }
        void show_mn()
        {
            cout<<"m="<<m<<"\n";
            cout<<"n="<<n<<"\n";
        }
};
int main()
{
    gamma g(6,14.3,24,46);
    g.show_x();
    g.show_y();
    g.show_mn();
    return 0;
}
```

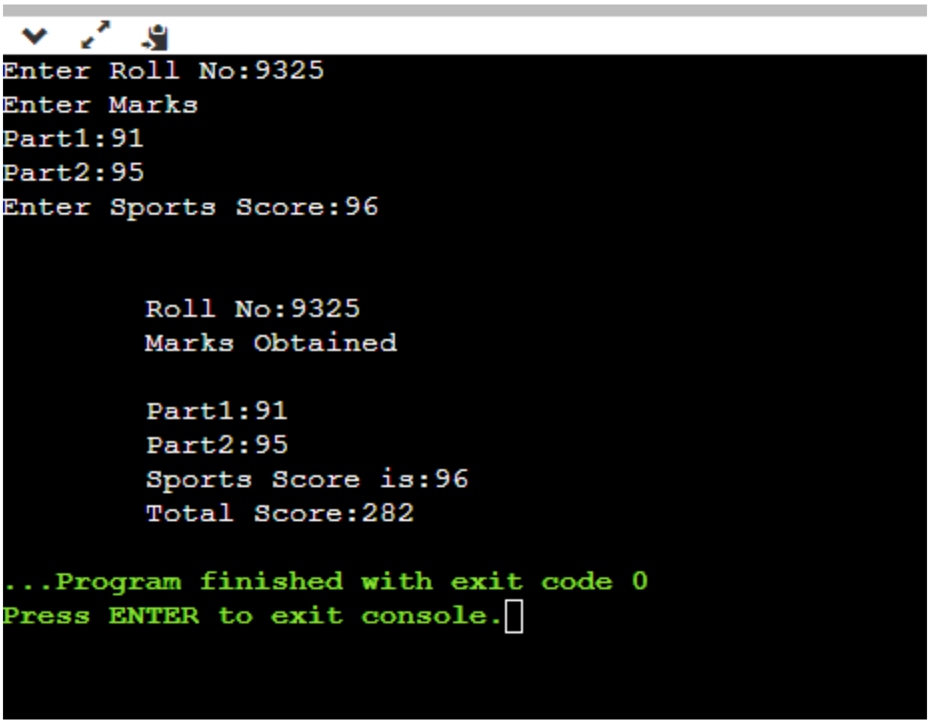
3. OUTPUT:

PROGRAM 5.1



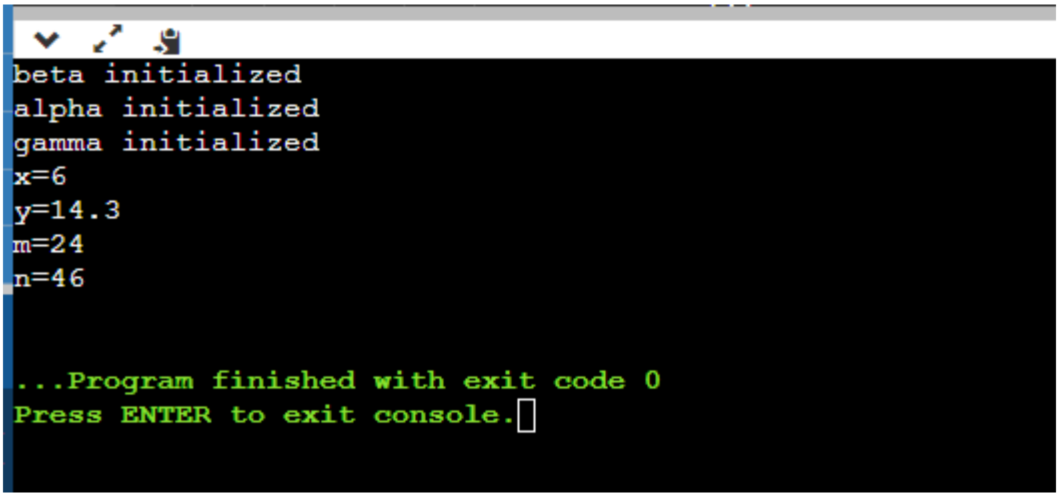
```
=====EDUCATION INSTITUTION DATABASE=====


Choose Category of Information
1)   Teachers
2)   Officer
3)   Typist
4)   Exit
Enter your choice:-1

=====TEACHERS INFORMATION=====


Choose your choice
1) Create
2) Display
3) Jump to Main Menu
Enter your choice:-1



Enter code :-9325
Enter name :-Karna Jaswanth
Enter Subject :-
Enter Publication :-Karna



Are you Interested in entering data
Enter y or n:-n

=====TEACHERS INFORMATION=====


Choose your choice
1) Create
```

PROGRAM 5.2:



```
Enter Roll No:9325
Enter Marks
Part1:91
Part2:95
Enter Sports Score:96


        Roll No:9325
        Marks Obtained

        Part1:91
        Part2:95
        Sports Score is:96
        Total Score:282

...Program finished with exit code 0
Press ENTER to exit console.
```

PROGRAM 5.3:



```
beta initialized
alpha initialized
gamma initialized
x=6
y=14.3
m=24
n=46


...Program finished with exit code 0
Press ENTER to exit console.
```

4. LEARNING OUTCOMES:

- Understand the concepts of object-oriented programming including programming process and compilation process.
- Apply different techniques to decompose a problem and programmed a solution with its sub modules.
- Analyze and explain the behavior of simple programs involving the programming addressed in the course.
- Implement and evaluate the programs using the syntax and semantics of object-oriented programming.
- Design the solution of real-world problems in order to determine that the program performs as expected.

EVALUATION COLUMN (To be filled by concerned faculty only)

Sr. No.	Parameters	Maximum Marks	Marks Obtained
1.	Worksheet Completion including writing learning objective/ Outcome	10	
2.	Post-Lab Quiz Result	5	
3.	Student engagement in Simulation/ Performance/ Pre-Lab Questions	5	
4.	Total Marks	20	