STUDENT REGISTRATION SYSTEM FOR A UNIVERSITY

Faculty of Applied Sciences Rajarata University of SriLanka

Department of Computing

L.G.R.J.Lindapitiya

Registration Number:ICT/1920/132

Index Number:5066

Contents

01.Introduction

This includes steps of making a Student Registration System for a University , using Object Oriented Programming Concepts. With the use of these concepts we can get a more clearer, reliable and less complex programs. Objects and Classes are the two main characteristics of Object Oriented Programming. Abstraction, Encapsulation, Inheritence

And Polymorphism and overloading are the other characteristics of Object Oriented Programming.This system allow students to register for courses ,academics to view registered students,management staff to view registered students and admins to create courses and create accounts. This University Management System had been created using c++ programing language.

03.Objects and Classes

Objects and Classes help to make a program more simple. It group materials in a program to simple groups namely, Objects and Classes.

```
class Student
    {
        private://Data Hiding
        int rollno;
        char name[20];
        char gender;
        int age;
        int std;
public://Data Abstraction
char *getName();
        {
    return name;
        }
int getRoll No();
        {
    return rollno;
        }
```

04.Abstraction

Abstraction is a process of defining the essential concepts while ignoring the inessential details. Providing only essential information to the outside world and hiding their background details to represent the needed information in program without presenting the details.

Example:

- *Class as an abstraction.*

This class Student gather all the information that a student could have to a single unit without representing them one by one each.

```
class Student
    {
        private://Data Hiding
         int rollno;
         char name[20];
         char gender;
         int age;
         int std;
public://Data Abstraction
char *getName();
          {
        return name;
          }
```

05.Encapsulation

This binds data and functions together that manipulate the data and keep both safe from outside interface and missues.This emits the concept of Data Hiding.

Example:

```
class Student
        {
            private://Data Hiding
             int rollno;
             char name[20];
             char gender;
             int age;
             int std;
        public://Data Abstraction
        char *getName();
                 {
              return name;
                 }
        int getRoll No();
                 {
              return rollno;
                 }
```

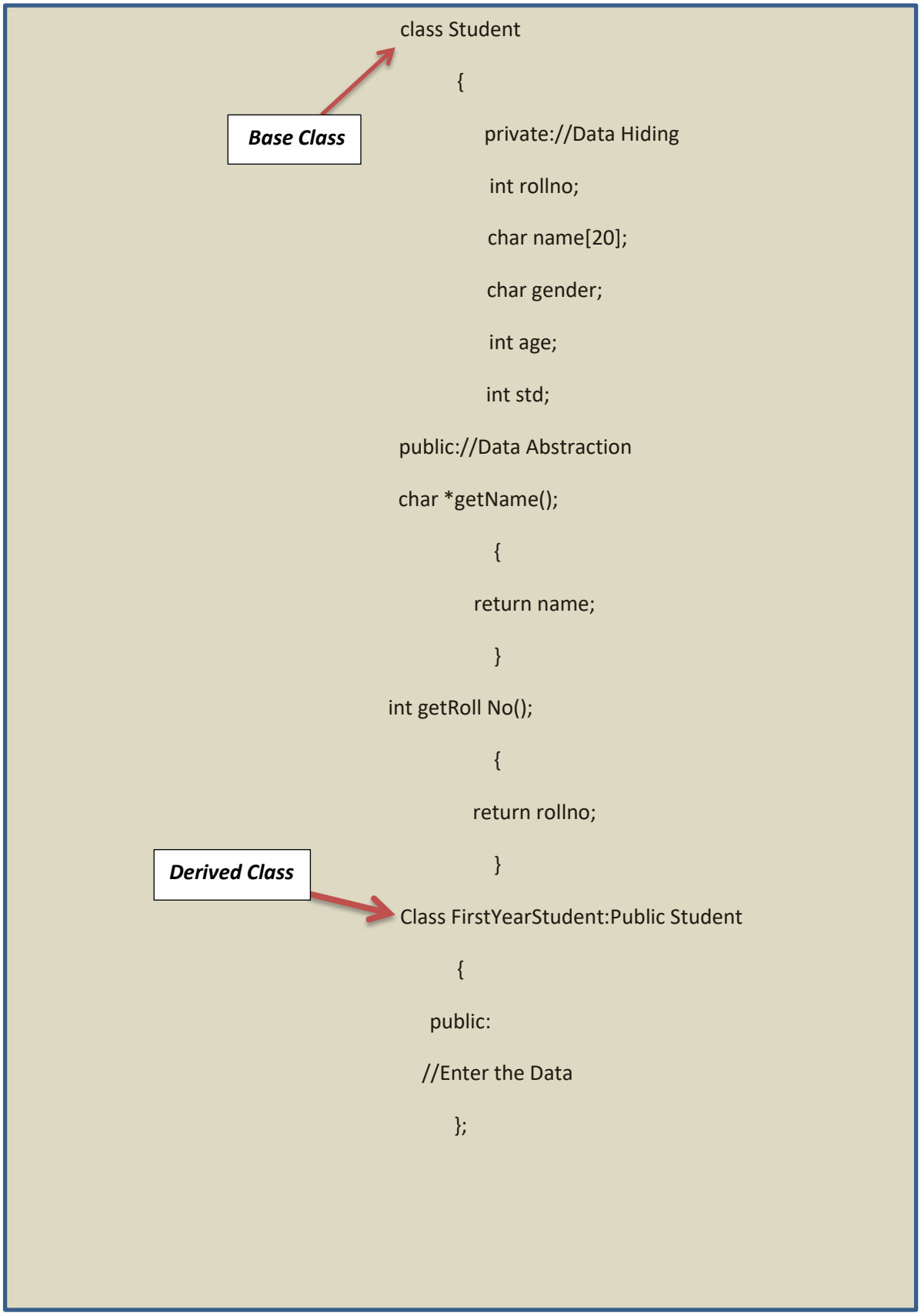
06.Inheritence

Inheritance is the process that classes in C++ can be extended by creating new classes which retain characteristics of the base class. Existing class is called Base class while the new class is called Derived classes.

Example:

```
class Student
        {
            private://Data Hiding
             int rollno;
             char name[20];
             char gender;
             int age;
             int std;
        public://Data Abstraction
        char *getName();
                 {
              return name;
                 }
       int getRoll No();
                 {
              return rollno;
                 }
```

Example:


Base Class
Derived Class
class Student
{
private://Data Hiding
int rollno;
char name[20];
char gender;
int age;
int std;
public://Data Abstraction
char *getName();
{
return name;
}
int getRoll No();
{
return rollno;
}
Class FirstYearStudent:Public Student
{
public:
//Enter the Data
};

07.Polymorphism and Overloading

C++ polymorphism means that a call to a member function will cause a different function to be executed depending on the type of object that invokes the function. There are two types in Polymorohism, namely Compiler Time and Run Time Polymorphisms. In C++ programming you can achieve compile time polymorphism in two ways as Overloading and Overriding.

Example:

Derived Class

Example:

```
void getData()//Member Functions
{
 cout<<"Stdent Details\n";
 cout<<"Roll No:"<<rollno<<endl;
 cin>>rollno;
 cout<<"Name:"<<name<<endl;
 cin>>name;
 cout<<"Gender:"<<gender<<endl;
 cin>>gender;
 cout<<"Age:"<<age<<endl;
 cin>>age;
 cout<<"Staddard:"<<std<<endl;
 cin>>std;
 }

void showData()//Member Functions
{
 cout<<"Stdent Details\n";
 cout<<"Roll No:"<<rollno<<endl;
 cout<<"Name:"<<name<<endl;
 cout<<"Gender:"<<gender<<endl;
 cout<<"Age:"<<age<<endl;
 cout<<"Staddard:"<<std<<endl;
```

08.Code

```
#include <iostream>
#include<fstream>
#include <string.h>
#include <iomanip>
#include<conio.h>

using namespace std;

void addRecord();//Fx to write data to file from memory

class Student {
private:
    int rollno;
    char name[20];
    char course[20];
    char gender[10];
    int age;
```

```
    int std;

public:
    char *getName() {
        return name;
    }

    int getRollNo() {
        return rollno;
    }

    void getData() {
        cout << "Student Details\n";

        cout << "Roll No:";
        cin >> rollno;

        cout << "Name:";
```

```
    cin >> name;

    cout << "Course:";
    cin >> course;


    cout << "Gender:";
    cin >> gender;

    cout << "Age:";
    cin >> age;

    cout << "Standard:";
    cin >> std;
}

void showData() {
    cout << "Student Details\n";
```

```
    cout << "Roll No:" << rollno << endl;
    cout << "Name:" << name << endl;
    cout << "Course:" << course << endl;
    cout << "Gender:" << gender << endl;
    cout << "Age:" << age << endl;
    cout << "Standard:" << std << endl;
}

void modifyData() {
    cout << "Roll No.:" << rollno << endl;
    cout << "Student Details\n";

    cout << "Roll No:";
    cin >> rollno;

    cout << "Name:";
    cin >> name;
```

```
    cout << "Course:";
    cin >> course;



    cout << "Gender:";
    cin >> gender;

    cout << "Age:";
    cin >> age;

    cout << "Standard:";
    cin >> std;
}

void heading() {
    cout.setf(ios::left);
    cout << setw(10) << "Roll No";
```

```
        cout << setw(20) << "Name";
        cout << setw(10) <<
"Course";
        cout << setw(10) <<
 "Gender";
        cout << setw(5) << "Age";
        cout << setw(5) << "Std";
        cout << endl;
    }

    void listData() {
        cout.setf(ios::left);
        cout << setw(10) << rollno;
        cout << setw(20) << name;
        cout << setw(10) <<
course ;
        cout << setw(10) <<
 gender;
```

```
        cout << setw(5) << age;
        cout << setw(5) << std;
        cout << endl;
    }
};
class mainclass{
    public:
void menu() {
    cout << "\nSTUDENT RECORDS\n";
    cout << "===============\n";
    cout << "0.Exit\n";
    cout << "1.Register New Student\n";
    cout << "2.Show All Students\n";
    cout << "3.Search by roll No:\n";
    cout << "4.Search by Name:\n";
    cout << "5.Modify Record:\n";
    cout << "6.Delete Record:\n";
```

```
    cout << "Enter your choice:\n";
}

void addRecord()
{
        ofstream fout;
        fout.open("Doc 1.xlsx",ios::out|ios::app|ios::binary);
        s.getData();
        fout.write((char*)&s,sizeof(Student));

        fout.close();
        cout<<"\nData Saved to file....\n";
}
};

int maintanor() {
    int n = 0, ch, rn, i, flag = 0, pos;
    Student s[10];
```

```
do {
    system("cls");
    menu();
    cin >> ch;
    switch (ch) {
    case 1:addRecord();
        //s[n].getData();
        //n++;
        //cout << "\nRecord saved successfully...\n";
        break;

    case 2:
        if (n > 0) {
            s[0].heading();
            for (i = 0; i < n; i++) {
                s[i].listData();
            }
        }
```

```
    else {
        cout << "\nNothing to display...\n";
    }
    break;

case 3:
    if (n > 0) {
        cout << "\nEnter roll No. to display:";
        cin >> rn;
        for(int i=0;i<n;i++)
                                {
                                    if(rn==s[i].getRollNo())
                                    {
                                        s[i].showData();
                                        flag++;
                                    }
                                    else
                                    {
```

```
                                        cout<<"\nNothing to
display...\n";
                                        }
                                        if(flag==0)
                                        {
                                                cout<<"\nNo such roll
no.found....\n";
                                        }

                                }
                        }
                break;

                case 0: exit(0);

                default:cout<<"\nInvalid Choise..\n";
        }

        cout<<"\n\nPress any key to continue.....\n";
```

```
            getch();

        }while(ch!=0);

}}
;

int main()
{
//Student s;
mainclass m;
m.maintanor();

return 0;
}
```