OOPS Lab Exam

Name: Tanmay Vig

Roll Num: 19BCS06

Source Code:

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

using namespace std;

class Employee {
   protected:
      string id, name;
   public:
      Employee(string ID, string NAME) {
         id = ID;
         name = NAME;
      }
      virtual void getDetails() {
      }
      string getID() {
         return id;
      }
};

class Teacher : public Employee {
   protected:
      string highestQual;
      string *subCodes;
```

```
    int numSubs;
  public:
    Teacher(string ID, string NAME, string QUAL, int NUMSUBS, string *SUBCODES) : Employee(ID, NAME) {
        highestQual = QUAL;
        numSubs = NUMSUBS;
        subCodes = new string[numSubs];
        for(int i=0;i<numSubs;i++) {
            subCodes[i] = SUBCODES[i];
        }
    }
    virtual void getDetails(){
        cout << "Name: " << name << endl << "ID: " << id << endl << "Highest Qualification: " << highestQual<< "Subjects Taught:" << endl;
    }
};

class HOD : public Teacher {
  private:
    string department;
  public:
    HOD(string ID, string NAME, string QUAL, int NUMSUBS, string *SUBCODES, string DEPT) : Teacher(ID, NAME, QUAL, NUMSUBS, SUBCODES) {
        department = DEPT;
    }
    void getDetails() {
        cout << "Name: " << name << endl << "ID: " << id << endl << "Highest Qualification: " << highestQual << endl << "Department: " << department << endl << "Subjects Taught:" << endl;
        for(int i=0;i<numSubs-1;i++) {
            cout << subCodes[i] << ", ";
```

```
        }
        cout << subCodes[numSubs-1] << endl;
    }
};

class Proctor : public Teacher {
  private:
    int yearsServed;
  public:
    Proctor(string ID, string NAME, string QUAL, int NUMSUBS, string *SUBCODES, int YEARS) :
Teacher(ID, NAME, QUAL, NUMSUBS, SUBCODES) {
        yearsServed = YEARS;
    }
    void getDetails() {
        cout << "Name: " << name << endl << "ID: " << id << endl << "Highest Qualification: " <<
highestQual << endl << "Number of years served: " << yearsServed << endl << "Subjects Taught:" <<
endl;
        for(int i=0;i<numSubs-1;i++) {
            cout << subCodes[i] << ", ";
        }
        cout << subCodes[numSubs-1] << endl;
    }
};

class Warden : public Teacher {
  private:
    int numHostelsAssigned;
  public:
    Warden(string ID, string NAME, string QUAL, int NUMSUBS, string *SUBCODES, int NUMHOS) :
Teacher(ID, NAME, QUAL, NUMSUBS, SUBCODES) {
```

```
        numHostelsAssigned = NUMHOS;
    }
    void getDetails() {
        cout << "Name: " << name << endl << "ID: " << id << endl << "Highest Qualification: " << highestQual << endl << "Number of hostels assigned: " << numHostelsAssigned << endl << "Subjects Taught:" << endl;
        for(int i=0;i<numSubs-1;i++) {
            cout << subCodes[i] << ", ";
        }
        cout << subCodes[numSubs-1] << endl;
    }
};

int main() {
  int numHod = -1, numProctor = -1, numWarden = -1;
  HOD *arrH[100];
  Proctor *arrP[100];
  Warden *arrW[100];

  while(true){
    int choice = 0, numYears = 0, numHostels = 0, numSubs = 0;
    string id, name, dept, highestQual;
    cout<<"\nPress 1 to add HOD, 2 to add Proctor, 3 to add Warden, 4 to show all HODs, 5 to show all Proctors, 6 to show all Wardens, 7 to find HOD, 8 to find proctor, 9 to find Warden, any num to exit\n";
    cin>>choice;
    switch(choice){
      case 1:
      {
        numHod++;
```

```
            cout<<"Enter Employee ID, Name, highest qualification, department and number of
subjects\n";
            cin.clear();
            cin.sync();
            getline(cin, id);
            getline(cin, name);
            getline(cin, highestQual);
            getline(cin, dept);
            cin>>numSubs;
            string *subs = new string[numSubs];
            cout<<"Enter subject codes\n";
            for(int i=0;i<numSubs;i++){
               string s;
               cin.clear();
               cin.sync();
               getline(cin, s);
               subs[i]=s;
            }
            arrH[numHod] = new HOD(id, name, highestQual, numSubs, subs, dept);
            break;
         }
         case 2:
         {
            numProctor++;
            cout<<"Enter Employee ID, Name, highest qualification, number of years served and number of
subjects\n";
            cin.clear();
            cin.sync();
            getline(cin, id);
```

```
            getline(cin, name);
            getline(cin, highestQual);
            cin>>numYears;
            cin>>numSubs;
            string *subs = new string[numSubs];
            cout<<"Enter subject codes\n";
            for(int i=0;i<numSubs;i++){
               string s;
               cin.clear();
               cin.sync();
               getline(cin, s);
               subs[i]=s;
            }
            arrP[numProctor] = new Proctor(id, name, highestQual, numSubs, subs, numYears);
            break;
         }
         case 3:
         {
            numWarden++;
            cout<<"Enter Employee ID, Name, highest qualification, number of hostels and number of
subjects\n";
            cin.clear();
            cin.sync();
            getline(cin, id);
            getline(cin, name);
            getline(cin, highestQual);
            cin>>numHostels;
            cin>>numSubs;
            string *subs = new string[numSubs];
```

```
    cout<<"Enter subject codes\n";
    for(int i=0;i<=numSubs;i++){
        string s;
        cin.clear();
        cin.sync();
        getline(cin, s);
        subs[i]=s;
    }
    arrW[numWarden] = new Warden(id, name, highestQual, numSubs, subs, numHostels);
    break;
}
case 4:
{
    cout << numHod;
    for(int i=0;i<=numHod;i++){
        (*arrH[i]).getDetails();
    }
    break;
}
case 5:
{
    for(int i=0;i<=numProctor;i++){
        (*arrP[i]).getDetails();
    }
    break;
}
case 6:
{
    for(int i=0;i<=numWarden;i++){
```

```
        (*arrW[i]).getDetails();
    }
    break;
}
case 7:
{
    int f=0;
    string ID;
    cout<<"Enter ID of HOD\n";
    cin.clear();
    cin.sync();
    getline(cin, ID);
    for(int i=0;i<=numHod;i++) {
        if(arrH[i]->getID() == ID) {
            f=1;
            arrH[i]->getDetails();
            break;
        }
    }
    if(f==0) {
        cout << "HOD with this ID does not exists\n";
    }
}
case 8:
{
    int f=0;
    string ID;
    cout<<"Enter ID of Proctor\n";
    cin.clear();
```

```
    cin.sync();
    getline(cin, ID);
    for(int i=0;i<=numProctor;i++) {
        if(arrP[i]->getID() == ID) {
            f=1;
            arrP[i]->getDetails();
            break;
        }
    }
    if(f==0) {
        cout << "HOD with this ID does not exists\n";
    }
}
case 9:
{
    int f=0;
    string ID;
    cout<<"Enter ID of Warden\n";
    cin.clear();
    cin.sync();
    getline(cin, ID);
    for(int i=0;i<=numWarden;i++) {
        if(arrW[i]->getID() == ID) {
            f=1;
            arrW[i]->getDetails();
            break;
        }
    }
    if(f==0) {
```

```
            cout << "HOD with this ID does not exists\n";
          }
        }
        default:
        {
          choice = 0;
          break;
        }
      }
      if(choice == 0) break;
    }

    return 0;
}
```

Output:

```
Wardens, 7 to find HOD, 8 to find proctor, 9 to find Warden, any num to exit
1
Enter Employee ID, Name, highest qualification, department and number of subjects
1
"Tanmay"
"PHD"
"Computer Engg"
2
Enter subject codes
CEN-102
CEN-301

Press 1 to add HOD, 2 to add Proctor, 3 to add Warden, 4 to show all HODs, 5 to show all Proctors, 6 to show
all Wardens, 7 to find HOD, 8 to find proctor, 9 to find Warden, any num to exit
2
Enter Employee ID, Name, highest qualification, number of years served and number of subjects
2
ALMAS
PHD
4
1
Enter subject codes
CEN-103
```

Yes, in this scenario we can implement run time polymorphism