# TASK 1:

#include<iostream>

#include<string>

using namespace std;

class employe

{

string id;

string name;

string department;

string const accNo;

string grade;

public:

employe() : accNo("0")

{

id = "0";

name = "0";

department = "0";

grade = "0";

}

employe(string id, string name, string department, string accNo, string grade) : accNo(accNo)

{

this->id = id;

this->name = name;

this->department = department;

this->grade = grade;

}

void input(string accNo)

{

cout << "Enter ID: ";

cin >> id;

cout << "Enter Name: ";

cin >> name;

cout << "Enter Department: ";

cin >> department;

cout << "Enter Grade: ";

cin >> grade;

accNo = accNo;

}

void display()

{

cout << "ID: " << id << endl;

cout << "Name: " << name << endl;

cout << "Department: " << department << endl;

cout << "Account NO: " << accNo << endl;

cout << "Grade: " << grade << endl;

}

~employe()

{

cout << "Destructor" << endl;

}

};

int main()

{

employe\* ptr = new employe[3];

string str;

for (int i = 0; i < 3; i++)

{

cout << "User: " << i + 1 << endl;

cout << "Enter Account NO: ";

cin >> str;

ptr[i].input(str);

cout << endl;

}

for (int i = 0; i < 3; i++)

{

cout << "User: " << i + 1 << endl;

ptr[i].display();

cout << endl;

}

for (int i = 0; i < 3; i++)

{

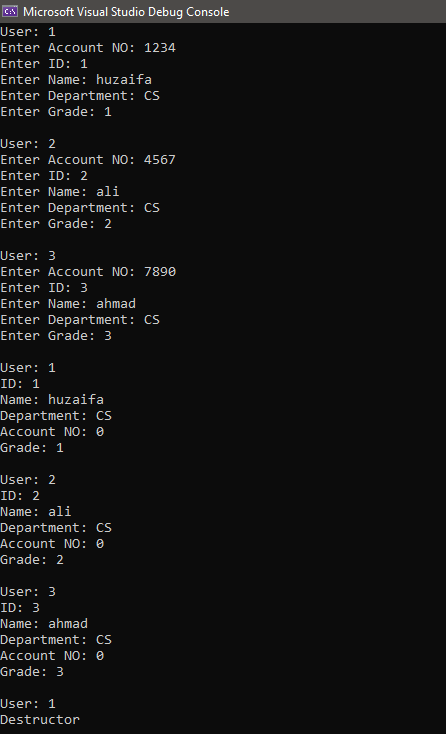
cout << "User: " << i + 1 << endl;

ptr[i].~employe();

}

}

# OUTPUT:



# 

# TASK 2:

**HEADER FILE:**

#pragma once

#include<iostream>

using namespace std;

class employe

{

private:

//Members

string name;

int id;

string department;

string position;

public:

//Default Constructor

employe();

//Parametrized Constructor

employe(string name, int id, string department, string position);

//Setters

void namesetter(string name);

void idsetter(int id);

void depsetter(string dep);

void possetter(string pos);

//Getters

string namegetter();

int idgetter();

string depgetter();

string posgetter();

};

**FUNCTION FILE:**

#include"Header.h"

#include<iostream>

using namespace std;

//Default Constructor

employe::employe()

{

name = "0";

id = 0;

department = "0";

position = "0";

}

//Parametrized Constructor

employe::employe(string name, int id, string dep, string pos)

{

this->name = name;

this->id = id;

this->department = dep;

this->position = pos;

}

//Setters

void employe::namesetter(string name)

{

this->name = name;

}

void employe::idsetter(int id)

{

this->id = id;

}

void employe::depsetter(string dep)

{

this->department = dep;

}

void employe::possetter(string pos)

{

this->position = pos;

}

//Getters

string employe::namegetter()

{

return name ;

}

int employe::idgetter()

{

return id;

}

string employe::depgetter()

{

return department;

}

string employe::posgetter()

{

return position;

}

**MAIN FILE:**

#include<iostream>

#include <iomanip>

#include"Header.h"

using namespace std;

int main()

{

employe e[3];

string name, dep, pos;

int id;

for (int i = 0; i < 3; i++)

{

cout << "For USER NO: " << i + 1 << endl;

cout << "Enter Name: ";

cin >> name;

cout << "Enter ID: " ;

cin >> id;

cout << "Enter Department: " ;

cin >> dep;

cout << "Enter Position: ";

cin >> pos;

cout << endl;

e[i].namesetter(name);

e[i].idsetter(id);

e[i].depsetter(dep);

e[i].possetter(pos);

}

cout << "NAME:" << setw(10) << "ID NUMBER:" << setw(10) << "DEPARTMENT:" << setw(10) << "POSITION:" << endl;

cout << e[0].namegetter() << setw(10) << e[0].idgetter() << setw(10) << e[0].depgetter() << setw(10) << e[0].posgetter() << endl;

cout << e[1].namegetter() << setw(10) << e[1].idgetter() << setw(10) << e[1].depgetter() << setw(10) << e[1].posgetter() << endl;

cout << e[2].namegetter() << setw(10) << e[2].idgetter() << setw(10) << e[2].depgetter() << setw(10) << e[2].posgetter() << endl;

}

# OUTPUT:

# 

# TASK 3:

**HEADER FILE:**

#pragma once

#include<iostream>

using namespace std;

class person

{

private:

string name;

const string dob;

const string cnic;

string mStatus;

public:

person(string name, string dob, string cnic, string mStatus);

const string dobgetter();

const string cnicgetter();

void Display();

};

**FUNCTION FILE:**

#include"Header.h"

#include<iostream>

using namespace std;

person::person(string name, string dob, string cnic, string mStatus) : dob(dob), cnic(cnic)

{

this->name = name;

this->mStatus = mStatus;

}

string const person::dobgetter()

{

return dob;

}

string const person::cnicgetter()

{

return cnic;

}

void person::Display()

{

cout << "DATA : " << endl;

cout << "Name :" << name << endl;

cout << "Date of birth : " << dobgetter() << endl;

cout << "CNIC : " << cnicgetter() << endl;

cout << "Marital Status: " << mStatus << endl;

}

**MAIN FILE:**

#include<iostream>

#include"Header.h"

using namespace std;

int main()

{

person p("Mr.X", "1st January 2001", "12345-6789101-1", "Married");

p.Display();

}

# OUTPUT:

