

|  |
| --- |
| [Assignment: 2] |
| Employee Management Application |
| [Object Oriented Programming (OOP) using (C++) |

***Under Supervisor:***

* Dr. Mahmoud A. Mahdi

***Faculty Of Computer Science and Artificial Intelligence ZagaZig University (FCIZU).***

***Programmed by:***

* Sherif Muhammad Muhammad Abdulsalam.
* Set Number: 1315
* Student Number: 20912021100785
* Group: A2
* Section: 13

|  |
| --- |
|  |

**Application Description:**

**To construct application using (C++ OOP), it was divided to subproblems (Classes) to create it alone such as Employee, (Staff Member), (Project), (Budget), (department) and(staff).**

**for each employee there are different data connected with the type of employee. the Staff Problem which has the ability to deal with all employees Who want to enroll the application and save them data. The staff class contain the main feathers of system like adding, editing, search, deleting or show all employees saved data.**

**There are other subproblems like (department) class and (create project) class.**

**The department class can add each employee to his department. While the Create project class is responsible for handling projects which are created by company.**

**Data Structures Used:**

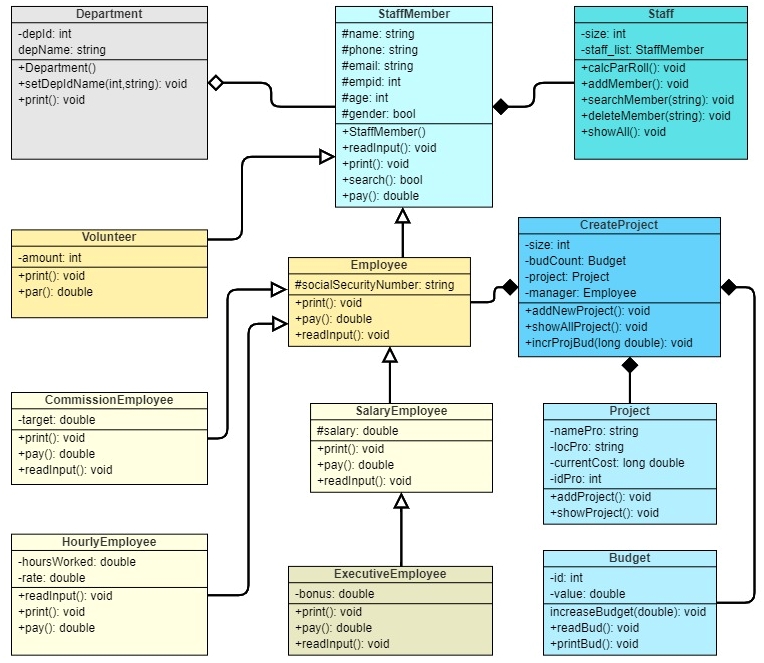
**Arrays:**

**Arrays has a variety of datatypes like built in datatypes (char, int, string…. etc.) and other are user defined data types (staff member, employee, project and other).**

**These data were collected to provide the best solution for collecting data together to be easily reached and used.**

**To give the best performance and save the memory we choose dynamic array to be used due to its performance and because it can fit automatically after insert or deleting objects or data from the array.**

UML Class Diagram



**Description for Classes and Functions Used:**

**Staff Member class**

**This class contain the main information foreach different type of employees (volunteer, salary, hourly, commission and executive employee) like name, id, phone, email, age, gender…. etc. from this class, there are some inherited classes which add more data to type of employees.**

**Department class**

**department class has the ability to add new department or print all departments and adding employees to their departments**

**Employee class**

**This inherited from staff member class and add the social security number to all employees except volunteers.**

**Volunteer class**

**Volunteer class add the amount of work to each volunteer only.**

**Salary, Hourly, Commission, Executive classes**

**These classes have the same data adding to the salary**

**to each different type of employees.**

**Staff class**

**Staff class take an array of staff member class information and can access on it by adding or deleting or search for any employee. It also can print all employee’s data and calculate the total payroll.**

**Project class**

**This class contain the main information foreach project like the project number(id), project location, project name and other info.**

**Budget class**

**Budget class contain the total budget for every project created and you can increase int by using the method (increase Budget).**

**Create Project**

**It takes an array of employees and array of project and array of budgets with the same index to add the project details which are manager from employees and project data from project class and the total budget from budget class.**

**Samples Output from Console Screen:**

User Interface

The Welcome Screen

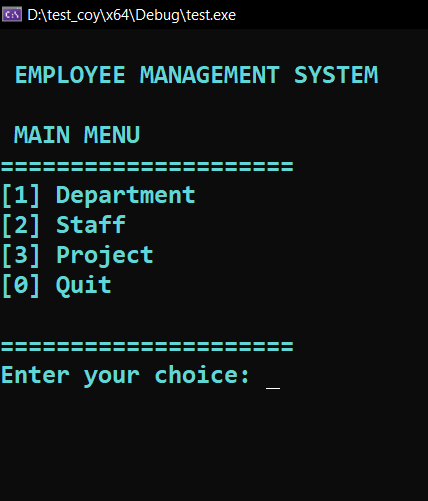
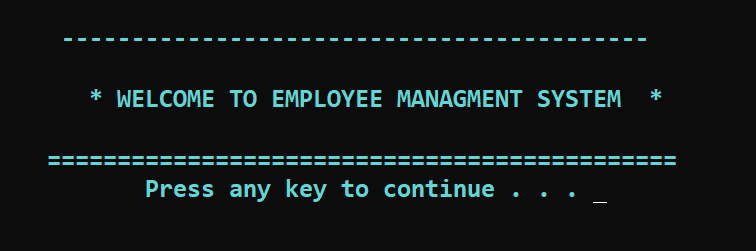


Figure no: 1

Department Menu

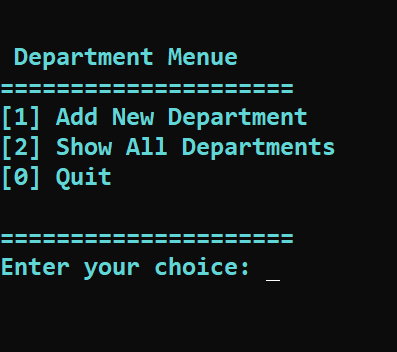


Figure no: 2

Add New Department

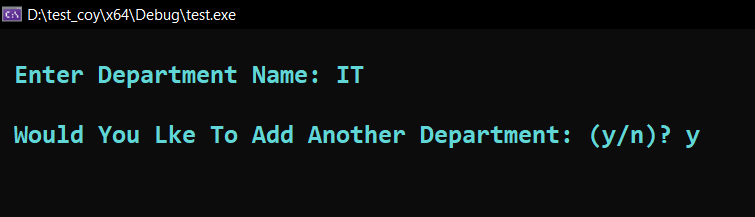
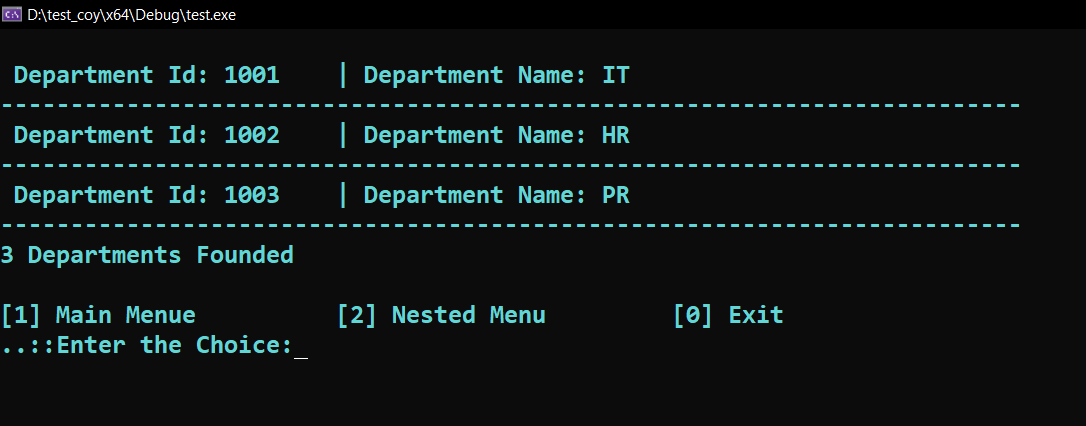


Figure no: 3

Figure no: 4

Show All Departments



Add Another Department

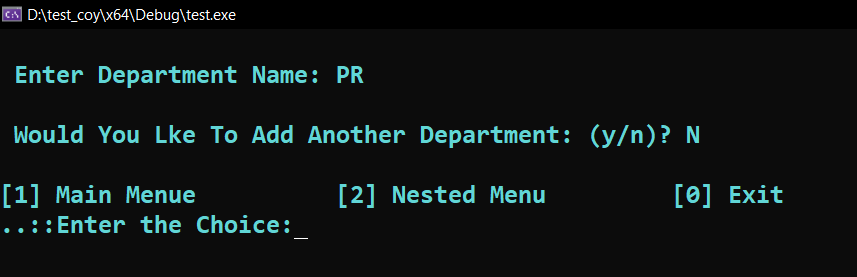


Figure no: 5

Figure no: 6

Member Type

Staff Menu

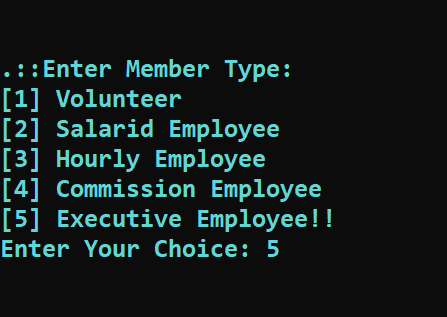
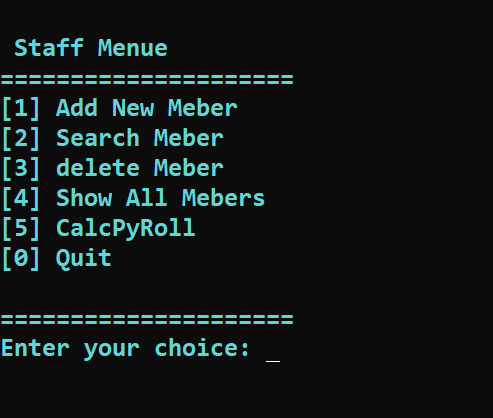


Figure no: 8

Figure no: 7

Add New Executive Member

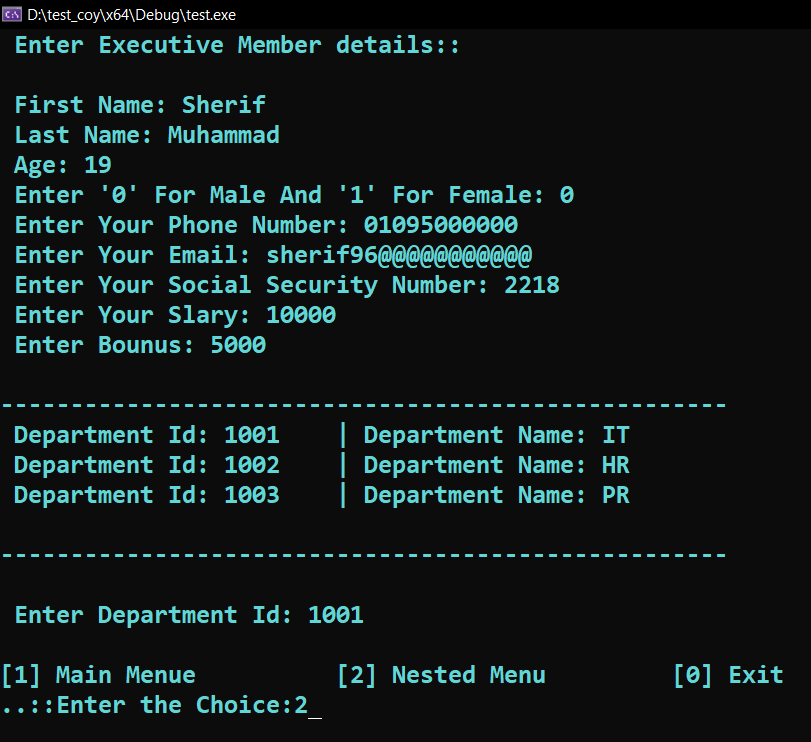
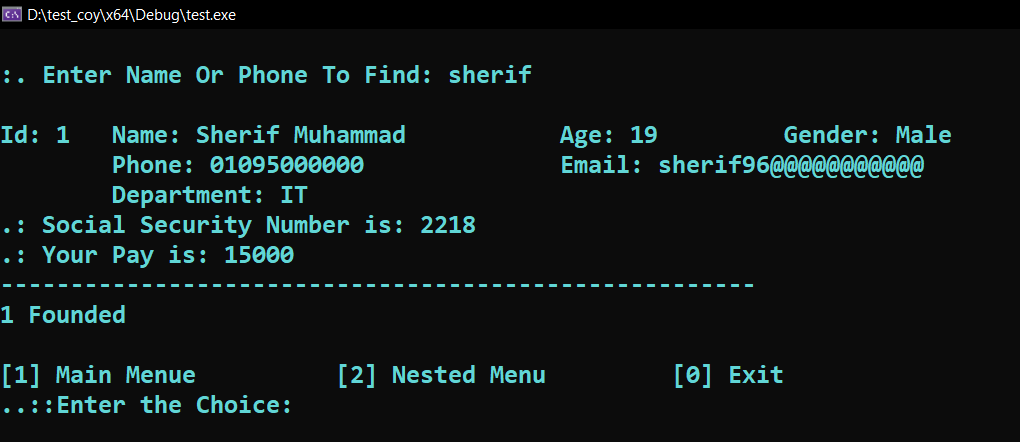


Figure no: 9



Search For Employees

Figure no: 10

Show All Employees

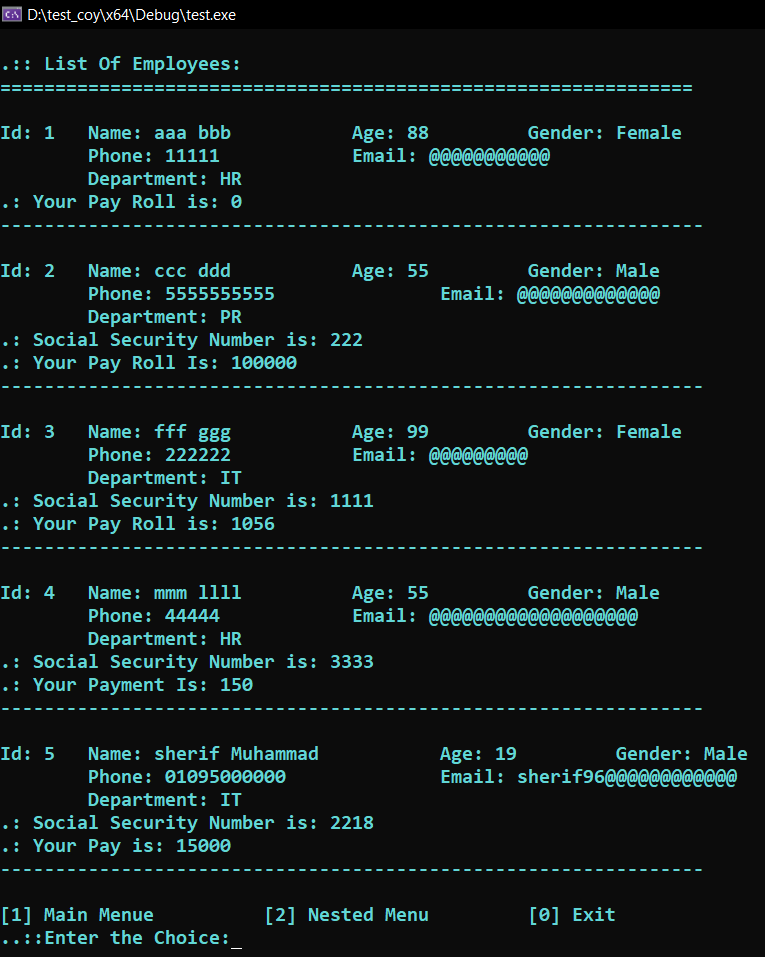


Figure no: 11

Delete An Employee

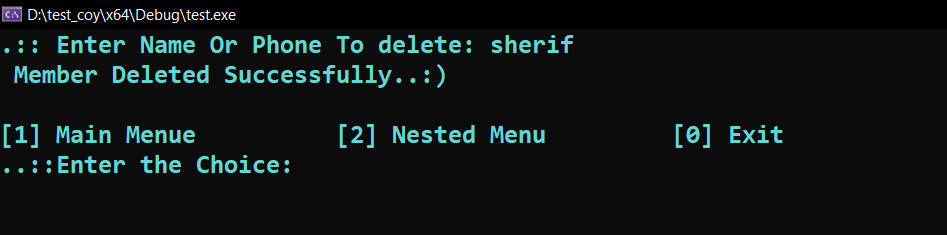


Figure no: 12

Show All Employees After Deleting 2 Employees

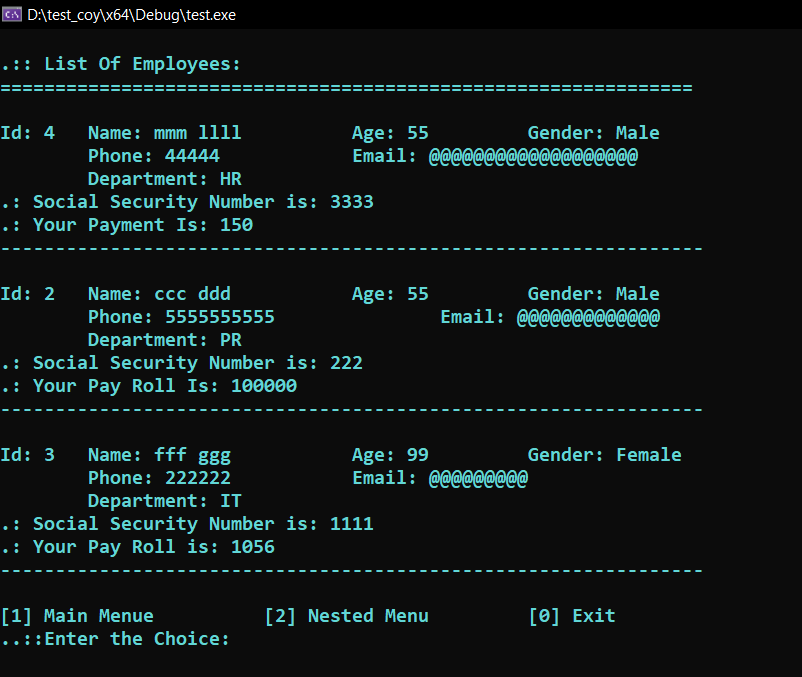


Figure no: 13

Calculate Payroll

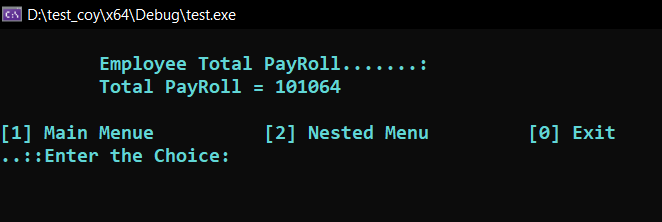


Figure no: 14

Add New Project

Project Menu

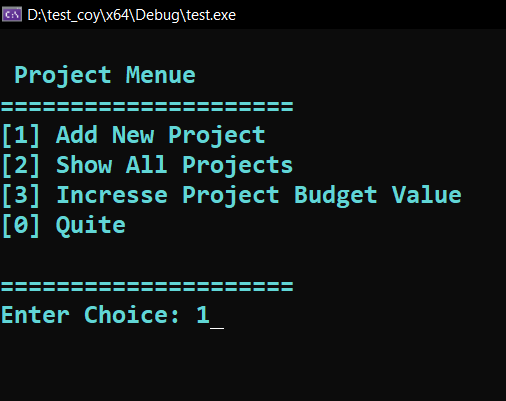
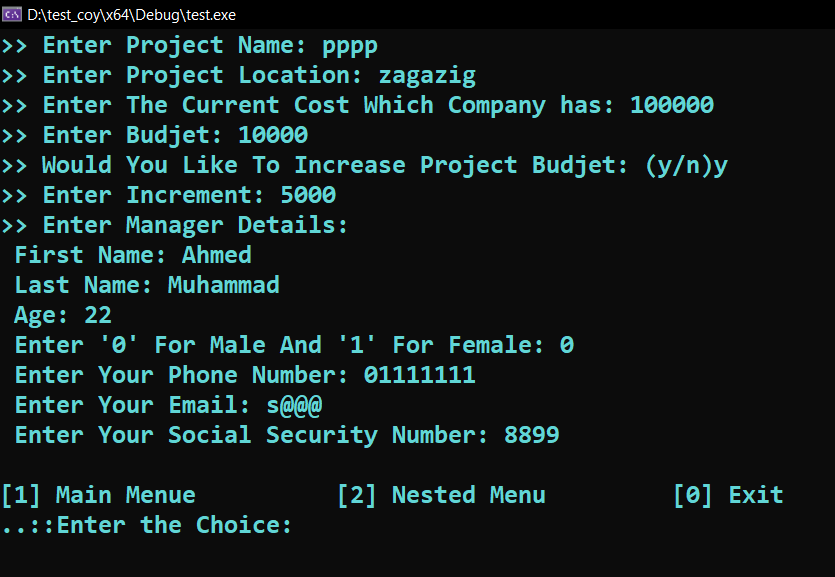


Figure no: 16

Figure no: 15

Show Project

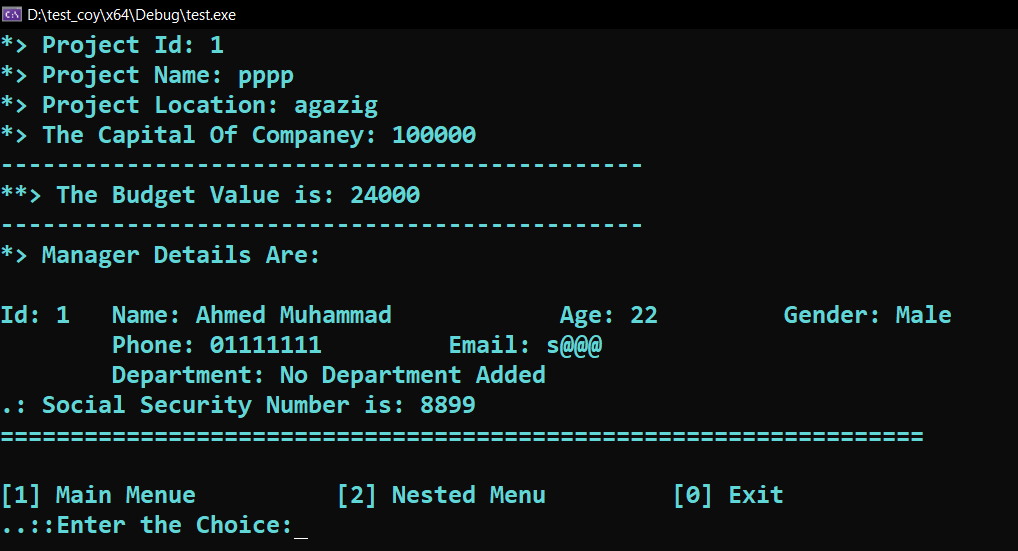


Figure no: 17

Increase Project Budget



Figure no: 18

Show Project After Increasing Budget

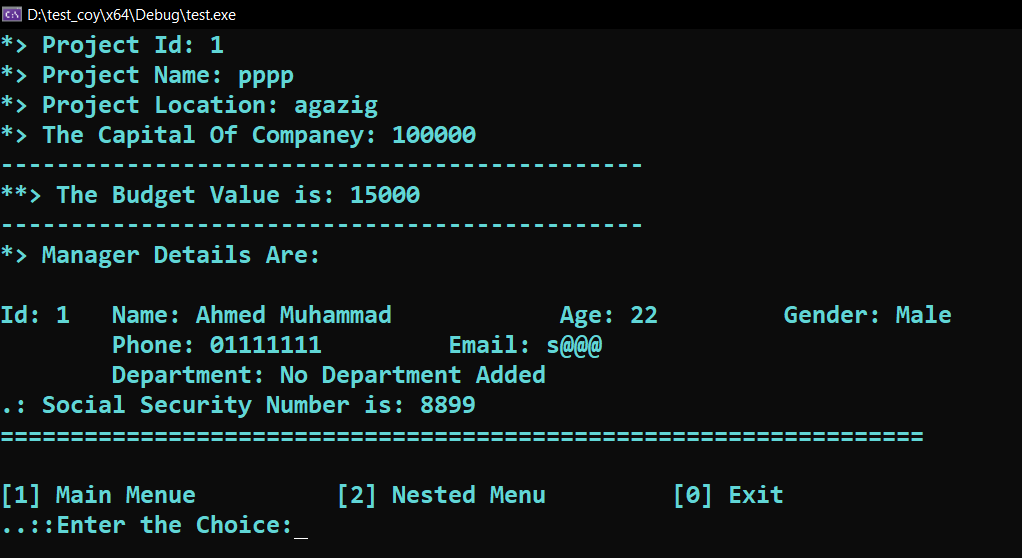


Figure no: 19

Exit Screen

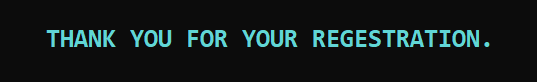


Figure no: 20

Main Source Code: \_\_source.cpp\_\_

#include<iostream>

#include"Staff.h"

#include"CreateProject.h"

#include<Windows.h>

#include"department.h"

using namespace std;

Staff staff(50); department\* dep = new department[100]; CreateProject proj(50);

auto depart\_counter = 0; auto c = -1; auto scc = -1;

int remain() {

int opt;

cout << "\n[1] Main Menue\t\t[2] Nested Menu\t\t[0] Exit\n"

<< "..::Enter the Choice:";

cin >> opt;

if (opt == 0) {

system("cls");

cout << "\n\n\n\t\t\tTHANK YOU FOR YOUR REGESTRATION." << endl;

return 1;

}

else if (opt == 1)

return -1;

}

void department\_handle() {

int choice = -1;

int id = 1000; int count = 0;

char ch = 'y';

string name;

while (choice != 0) {

system("cls");

cout << "\n\n Department Menue";

cout << "\n=====================\n" << "[1] Add New Department\n"

<< "[2] Show All Departments\n"

<< "[0] Quit\n" << "\n=====================\n" << "Enter your choice: ";

cin >> choice;

if (choice == 0) break;

else if (choice == 1) {

do {system("cls");

cout << "\n Enter Department Name: "; cin >> name;

dep[count++].setDepIdName(++id, name);depart\_counter++;

cout << "\n Would You Lke To Add Another Department: (y/n)? "; cin >> ch;

} while (ch == 'y' || ch == 'Y');

int opt;

cout << "\n[1] Main Menue\t\t[2] Nested Menu\t\t[0] Exit\n"

<< "..::Enter the Choice:";

cin >> opt;

switch (opt)

{

case 1 || 0; return;

case 2: break;

default:

break;

}

}

**Copy Of C++ Code:**

else if (choice == 2) {

system("cls"); cout << endl;

for (size\_t i = 0; i < count; i++)

{

dep[i].print();

cout << "---------------------------------------------------------------\n";

}

if (depart\_counter == 0) cout << "\n No Department Found...!!):\n";

else cout << depart\_counter << " Departments Founded\n";

int opt;

cout << "\n[1] Main Menue\t\t[2] Nested Menu\t\t[0] Exit\n"

<< "..::Enter the Choice:"; cin >> opt;

switch (opt)

{

case 1 || 0: return;

case 2: break;

default: break;

}

}

else {

cout << " Invalid Chioce!!";

Sleep(3000);

}

}

}

int main() {

system("cls");

system("Color 0B");

cout << "\n\n\n\n\n\n\n\n\n\n\n\n\t\t\t\t ------------------------------------------"

<< "\n\n\t\t\t\t \* WELCOME TO EMPLOYEE MANAGMENT SYSTEM \*"

<< "\n\n\t\t\t\t ============================================="; cout << "\n\t\t\t\t\t "; system("pause");

while (c != 0) {

system("cls"); system("Color 0B");

cout << "\n EMPLOYEE MANAGEMENT SYSTEM"; cout << "\n\n MAIN MENU";

cout << "\n=====================\n";

cout << "[1] Department\n" << "[2] Staff\n" << "[3] Project\n" << "[0] Quit\n";

cout << "\n=====================\n";

cout << "Enter your choice: "; cin >> c; system("cls");

switch (c) {

case 1:

department\_handle();

break;

case 2:

while (scc != 0) {

system("cls"); cout << "\n\n Staff Menue";

cout << "\n=====================\n" << "[1] Add New Meber\n"

<< "[2] Search Meber\n" << "[3] delete Meber\n"

<< "[4] Show All Mebers\n" << "[5] CalcPyRoll\n"

<< "[0] Quit\n" << "\n=====================\n"

<< "Enter your choice: "; cin >> scc; system("cls");

if (scc == 1) {

staff.AddNewMember(dep, depart\_counter);

int rem = remain(); if (rem == 1) exit(0);

else if (rem == -1)

break;

}

else if (scc == 4) {

staff.printAllMebers();

int rem = remain();

if (rem == 1) exit(0); else if (rem == -1) break;

}

else if (scc == 2) {

staff.SearchMember();

int rem = remain();

if (rem == 1) exit(0); else if (rem == -1) break;

}

else if (scc == 3) {

staff.deleteMember();

int rem = remain();

if (rem == 1) exit(0);

else if (rem == -1) break;

}

else if (scc == 5) {

system("cls");

staff.calcPayRoll();

int rem = remain();

if (rem == 1) exit(0);

else if (rem == -1) break;

}

}

break;

case 3:

while (scc != 0) {

system("cls");

cout << "\n Project Menue";

cout << "\n=====================\n";

cout << "[1] Add New Project \n" << "[2] Show All Projects \n"

<< "[3] Incresse Project Budget Value \n" << "[0] Quite \n";

cout << "\n=====================\n";

cout << "Enter Choice: "; cin >> scc;

if (scc == 1) { system("cls");

proj.addNewProject();

int rem = remain(); if (rem == 1) exit(0); else if (rem == -1) break; }

else if (scc == 2) { system("cls");

proj.showAllProject();

int rem = remain(); if (rem == 1) exit(0); else if (rem == -1) break; }

else if (scc == 3) {system("cls");int \_id;

cout << " Enter Project Id: "; cin >> \_id;

proj.incrProjBud(\_id); int rem = remain(); if (rem == 1) exit(0);

else if (rem == -1) break;

}

}

break;

default:

cout << " \n\n\n\tInvalid Chioce!!\n\tWait for seconds And Enroll Again\n";

Sleep(3000);

break;

}

}

system("cls");

cout << "\n\n\n\t\t\tThank you." << endl << endl;

exit(0);

return 0;

}

Staff Source Code: \_\_Staff.h\_\_

#pragma once

#include"StaffMember.h"

#include"department.h"

class Staff{

int size; int count\_total;

StaffMember\*\* staff\_listt;

public:

Staff(int); ~Staff();

void AddNewMember(department\*,int);

void SearchMember();void deleteMember();

void printAllMebers();void calcPayRoll();

};

Staff Source Code: \_\_Staff.cpp\_\_

#include "Staff.h"

#include"StaffMember.h"

#include"Commission.h"

#include"S\_Employee.h"

#include"Executive.h"

#include"H\_Employee.h"

#include"Volunteer.h"

#include<iostream>

using namespace std;

Staff::Staff(int count) :count\_total(0) { size = count; staff\_listt = new StaffMember\*[size];}

Staff::~Staff(){

for (size\_t i = 0; i < size; ++i) {delete staff\_listt[i];}

delete[]staff\_listt;

}

Department Source Code: \_\_Department.cpp\_\_

#include "department.h"

#include<iostream>

department::department():depId(0),depName("No Department Added") {}

void department::setDepIdName(int id, string name) { depId = id; depName = name;}

string department::getName() const {return depName;}

void department::print() const{

cout << " Department Id: " << depId << "\t| Department Name: "

<< depName << endl;}

Department Source Code: \_\_Department.h\_\_

#pragma once

#include<string>

using namespace std;

class department{

int depId;string depName;

public:

department();void setDepIdName(int, string);

string getName() const;void print() const;};

void Staff::AddNewMember(department\*depList,int size) {

int type = -1; int c;

cout << "\n\n.::Enter Member Type: \n"; cout << "[1] Volunteer\n[2] Salarid Employee\n"

"[3] Hourly Employee \n[4] Commission Employee\n[5] Executive Employee!!";

cout << "\nEnter Your Choice: "; cin >> type;

bool flag = true;

if (type == 1) { system("cls"); staff\_listt[count\_total] = new Volunteer;}

else if (type == 2) {system("cls");

cout << " Enter Salaried Employee Member details:: " << endl;

staff\_listt[count\_total] = new S\_Employee;}

else if (type == 3) { system("cls"); staff\_listt[count\_total] = new H\_Employee;}

else if (type == 4) {system("cls");staff\_listt[count\_total] = new Commission;}

else if (type == 5) {system("cls");staff\_listt[count\_total] = new Executive;}

else flag = false;

if (flag) {

staff\_listt[count\_total]->setEmpId(count\_total + 1);

staff\_listt[count\_total]->readInput();

cout << "\n----------------------------------------------------\n";

for (size\_t i = 0; i < size; i++) depList[i].print();

cout << "\n----------------------------------------------------\n";

cout << "\n Enter Department Id: "; cin >> c;

staff\_listt[count\_total]->setDepart(depList[c - 1001]);

++count\_total;

}

}

void Staff::SearchMember() {

string keySer;int foundMemeber = 0;

cout << "\n:. Enter Name Or Phone To Find: "; cin >> keySer;

for (size\_t i = 0; i < count\_total; ++i) {

if (staff\_listt[i]->search(keySer)) {staff\_listt[i]->print();

cout << "------------------------------------------------------\n";

++foundMemeber;

}

}

if (foundMemeber == 0) cout << "\n\nNo Member Found!!\n";

else cout << foundMemeber << " Founded\n";

}

void Staff::printAllMebers()

{

if (count\_total == 0) { cout << " \n\nNo Employees Found!!!\n";}

else { cout << "\n.:: List Of Employees: \n";

cout << "===============================================================\n";

for (int i = 0; i < count\_total; ++i){

staff\_listt[i]->print();

cout << "----------------------------------------------------------\n";

}

}

}

void Staff::calcPayRoll()

{

cout << "\n\t Employee Total PayRoll.......:\n";

double \_total = 0;

for (size\_t i = 0; i < count\_total; ++i)

{

\_total += staff\_listt[i]->pay();

}

cout << "\t Total PayRoll = " << \_total << endl;

}

void Staff::deleteMember(){

string keySer;bool delet = false;

cout << ".:: Enter Name Or Phone To delete: "; cin >> keySer;

for (size\_t i = 0; i < count\_total; i++)

{

if (staff\_listt[i]->search(keySer)) {

if (i == count\_total - 1) count\_total--;

else {

staff\_listt[i] = staff\_listt[count\_total - 1];

count\_total--;

}

}

delet = true;break;

}

if (!delet) cout << "\n\nNo Member Found To Delete..:(\n";

else cout << " Member Deleted Successfully..:)\n";

}

Staff Member Source Code: \_\_staffMember.cpp\_\_

#include "StaffMember.h"

#include<iostream>

using namespace std;

StaffMember::StaffMember():fName("No Name"),age(0),gender("No Gender Added"){

setName(fName, lName);setEmpId(empid);setAge(age);setGender(gender);}

StaffMember::~StaffMember(){}

void StaffMember::setName(string fname, string lname) {fName = fname;lName = lname;}

void StaffMember::setEmpId(int id) {empid = id;}

int StaffMember::getId() { return empid; }

void StaffMember::setAge(int age) { this->age = age;}

void StaffMember::setGender(bool gend) { gender = gend;}

bool StaffMember::search(string key){

if (fName.compare(key) == 0 || lName.compare(key) == 0 || phone.compare(key) || email.compare(key)) return true;

return false;

}

Staff Member Source Code: \_\_staffMember.h\_\_

#pragma once

#include<string>

#include"department.h"

using namespace std;

class StaffMember{

protected:

string fName, lName; int empid,age;

bool gender; string phone;string email;

department depart;

public:

StaffMember(); ~StaffMember();

void setName(string ,string); void setEmpId(int);

void setAge(int); int getId(); void setGender(bool);

bool search(string); virtual void readInput();

virtual void print(); virtual double pay() = 0;

void setDepart(department);

};

void StaffMember::readInput()

{

cout << " \n First Name: ";cin.ignore(); cin >> fName;

cout << " Last Name: "; cin >> lName;

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

cout << " Enter '0' For Male And '1' For Female: ";cin >> gender;

cout << " Enter Your Phone Number: "; cin >> phone;

cout << " Enter Your Email: "; cin >> email;

}

void StaffMember::print()

{

cout << "\nId: " << empid << "\tName: " << fName << " " << lName;

if (age > 0) {

cout << "\t\tAge: " << age;

}

if (gender == 0) cout << " \tGender: Male";

else cout << " \tGender: Female";

cout << "\n\tPhone: " << phone;

cout << "\t\tEmail: " << email;

cout << "\n\tDepartment: " << depart->getName() << endl;

}

void StaffMember::setDepart(department \*depart) {

this->depart = depart;

}

Employee Source Code: \_\_Employee.cpp\_\_

#include "Employee.h"

#include<iostream>

using namespace std;

Employee::Employee(){setsocialSecurutyNumber(socialSecurutyNumber);}

void Employee::setsocialSecurutyNumber(string soc) {socialSecurutyNumber = soc;}

void Employee::readInput(){ StaffMember::readInput();

cout << " Enter Your Social Security Number: ";cin >> socialSecurutyNumber;}

void Employee::print()

{StaffMember::print();

cout << ".: Social Security Number is: " << socialSecurutyNumber << endl;}

double Employee::pay() {return 0.0;}

Employee Source Code: \_\_Employee.h\_\_

#pragma once

#include"StaffMember.h"

#include<string>

using namespace std;

class Employee :public StaffMember{

protected:

string socialSecurutyNumber;

public:

Employee();

void setsocialSecurutyNumber(string);

virtual void readInput();

virtual void print(); virtual double pay();

};

Volunteer Source Code: \_\_Volunteer.cpp\_\_

#include "Volunteer.h"

#include<iostream>

void Volunteer::setAmount(int a){amount = a;}

void Volunteer::readInput(){cout << " Enter Volunteer Member details:: " << endl;

StaffMember::readInput();cout << " Enter Amount: "; cin >> amount;

}

void Volunteer::print(){StaffMember::print();std::cout << ".: Your Pay Roll is: 0\n";}

double Volunteer::pay(){return amount;}

Volunterr Source Code: \_\_Volunteer.h\_\_

#pragma once

#include"StaffMember.h"

class Volunteer:public StaffMember{

int amount = 0;

public:

void setAmount(int);

void readInput();

void print();

double pay();

};

S\_Employee Source Code: \_\_S\_Employee.h\_\_

#pragma once

#include"Employee.h"

class S\_Employee:public Employee{

protected:

double salary;

public:

S\_Employee();

void setSalary(double);

virtual void readInput();

virtual void print();

virtual double pay();

};

Employee Source Code: \_\_S\_Employee.cp\_\_

#include "S\_Employee.h"

#include<iostream>

#include<string>

using namespace std;

S\_Employee::S\_Employee() {setSalary(salary);}

void S\_Employee::setSalary(double s){ salary = s;}

void S\_Employee::readInput() { Employee::readInput();

cout << " Enter Your Slary: ";cin >> salary;}

void S\_Employee::print(){Employee::print();

cout << ".: Your Pay Roll Is: " << this->pay() << endl;}

double S\_Employee::pay() {return salary;}

Commission Source Code: \_\_Commission.h\_\_

#include "Commission.h"

#include<iostream>

using namespace std;

void Commission::setTarget(double tar){target = tar;}

Commission::Commission(){target = 0;}

void Commission::readInput()

{double tarr;cout << " Enter Commission Member details:: " << endl;

Employee::readInput();cout << " Enter Your Target: "; cin >> tarr;

setTarget(tarr);}

void Commission::print()

{Employee::print();cout << ".: Your Payment Is: " << target << endl;}

double Commission::pay(){return 0.05 \* target;}

Commission Source Code: \_\_Commission.h\_\_

#pragma once

#include"Employee.h"

class Commission:public Employee{

double target;

public:

Commission();

void setTarget(double);

void readInput();

void print();double pay();

};

H\_Employee Source Code: \_\_H\_Employee.cpp\_\_

#include "H\_Employee.h"

#include<iostream>

H\_Employee::H\_Employee(){setRateWork(rate, hoursWorked);}

void H\_Employee::setRateWork(double rate, double work){

this->rate = rate;hoursWorked = work;}

double H\_Employee::pay(){return rate\*hoursWorked;}

void H\_Employee::readInput(){

cout << " Enter Hourly Employee Member details:: " << endl;

Employee::readInput();cout << " Enter Rate: "; cin >> rate;

cout << " Enter Hours Worked: "; cin >> hoursWorked;}

void H\_Employee::print(){Employee::print();

cout << ".: Your Payment is: " << rate \* hoursWorked << endl;}

H\_Employee Source Code: \_\_H\_Employee.h\_\_

#pragma once

#include"Employee.h"

class H\_Employee:public Employee{

double rate, hoursWorked;

public:

H\_Employee();

void setRateWork(double, double);

double pay();void readInput();void print();

};

Executive Source Code: \_\_ Executive.cpp\_\_

#include "Executive.h"

#include<iostream>

Executive::Executive(){

addBounus(bounus);

}

void Executive::addBounus(double BB) {

bounus = BB;

}

double Executive::pay(){return bounus + salary;}

void Executive::readInput(){

cout << " Enter Executive Member details:: " << endl;

S\_Employee::readInput(); cout << " Enter Bounus: "; cin >> bounus;

}

void Executive::print(){

Employee::print();cout << ".: Your Pay is: " << pay() << endl;

}

Executive Source Code: \_\_ Executive.h\_\_

#pragma once

#include"S\_Employee.h"

class Executive:public S\_Employee

{

double bounus;

public:

Executive();

void addBounus(double);

double pay();

void readInput();

void print();

};

CreateProject Source Code: \_\_ CreateProject.h\_\_

#pragma once

#include"Project.h"

#include"Budget.h"

#include"Employee.h"

class CreateProject

{

Budget \*budCount;

int size;

int count;

Project\* project;

Employee\* manager;

public:

CreateProject(int);

~CreateProject();

void addNewProject();

void showAllProject();

void incrProjBud(long double idddd);

};

CreateProject Source Code: \_\_ CreateProject.cpp\_\_

#include "CreateProject.h"

#include<iostream>

#include<string>

CreateProject::CreateProject(int amount) {count = 0;size = amount;

project = new Project[size];

manager = new Employee[size];

budCount = new Budget[size];

}

void CreateProject::addNewProject()

{

Project\* newProject = new Project;

Employee\* newmanger = new Employee;

Budget\* newBudget = new Budget;

newProject->setProId(count + 1);

newmanger->setEmpId(count + 1);

budCount->setBudId(count + 1);

newProject->addProject();

newBudget->readBud();

cout << ">> Enter Manager Details:";

newmanger->readInput();

project[count] = \*newProject;

manager[count] = \*newmanger;

budCount[count] = \*newBudget;

++count;

}

void CreateProject::showAllProject()

{

bool pro = false;

for (size\_t i = 0; i < count; i++)

{

pro = true;

project[i].showProject();

cout << "----------------------------------------------\n";

budCount[i].printBud();

cout << "----------------------------------------------";

cout << "\n\*> Manager Details Are: \n";

manager[i].print();

cout << "==================================================================\n";

}

if (!pro) cout << "\n\t No ProJect Found!!\n";

}

void CreateProject::incrProjBud(long double idddd){

bool \_found = false;

int increase;

for (size\_t i = 0; i < count; i++)

{

if (project[i].getProId() == idddd) {

cout << " \n>> Enter Budget Increment: "; cin >> increase;

budCount[i].increaseBud(increase);

\_found = true;break;

}

}

if (!\_found)cout << "\n\t Project Not Found!!\n";

}

CreateProject::~CreateProject(){delete[]project;delete[]manager;delete[]budCount;}

Project Source Code: \_\_ Project.cpp\_\_

#include "Project.h"

#include<iostream>

using namespace std;

void Project::setProId(int id){proId = id;}

int Project::getProId(){ return proId;}

void Project::setProName(string name){namePro = name;}

void Project::setProLoc(string loc){locPro = loc;}

void Project::addProject()

{

cout << ">> Enter Project Name: ";

cin.ignore(); getline(cin, namePro);

cout << ">> Enter Project Location: ";

cin.ignore(); getline(cin, locPro);

cout << ">> Enter The Current Cost Which Company has: ";

cin >> costCurr;

}

void Project::showProject()

{

cout << "\*> Project Id: " << proId << endl;

cout << "\*> Project Name: " << namePro << endl;

cout << "\*> Project Location: " << locPro << endl;

cout << "\*> The Capital Of Companey: " << costCurr << endl;

}

Project Source Code: \_\_ Project.h\_\_

#pragma once

#include"Budget.h"

#include"Employee.h"

#include"string"

class Project

{

int proId;

long double costCurr;

string namePro, locPro;

public:

void setProId(int id);

int getProId();

void setProName(string name);

void setProLoc(string loc);

virtual void addProject();

virtual void showProject();

};

Bydget Source Code: \_\_ Budget.cpp\_\_

#include "Budget.h"

#include<iostream>

using namespace std;

void Budget::setBud(double value)

{

this->value = value;

}

void Budget::setBudId(int bbb)

{

budid = bbb;

}

int Budget::getBudId() const

{return budid;}

void Budget::increaseBud(double incr)

{value += incr;}

double Budget::getBud() const

{return value;}

void Budget::readBud()

{

cout << ">> Enter Budjet: ";

cin >> value;

char aaa = 'y';

cout << ">> Would You Like To Increase Project Budjet: (y/n)";

cin >> aaa;

if (aaa == 'y' || aaa == 'Y') {

int inc;

cout << ">> Enter Increment: ";

cin >> inc;

this->increaseBud(inc);

}

}

void Budget::printBud() const

{

cout << "\*\*> The Budget Value is: " << this->getBud() << endl;

}

Bydget Source Code: \_\_ Budget.h\_\_

#pragma once

class Budget {

int budid = 0;

double value = 0;

public:

void setBud(double value);

void setBudId(int bbb);

int getBudId() const ;

void increaseBud(double incr);

double getBud() const;

void readBud();

void printBud() const;

};