

Object Oriented Programming

# Practice Lab 1

Name: Muhammad Hussain Javed

Class: BCY-2B

Roll Number: 21K-3584

Task 1

**#include <iostream>**

**#include <string>**

**// The following program consist of a class StudentSociety. The main purpose of this program is**

**// to demostrate how a Student society would fucntion if it was a program.**

**using namespace std;**

**class StudentSociety{**

**// ints declared with "i", strings with "s" etc.**

**string ssociety\_name, sfaculty\_head, spresident, sgeneral\_secretary;**

**string sevents[100], vote[2];**

**int ibudget, icnt;**

**public:**

**StudentSociety()**

**{**

**//The following constucter is a default constructer and takes user input.**

**cout << "Input the Society name: ";**

**getline(cin, ssociety\_name);**

**cout << "Input the name of the Faculty Head: ";**

**getline(cin, sfaculty\_head);**

**cout << "Input the name of the President: ";**

**getline(cin, spresident);**

**cout << "Input the name of the General Secretary: ";**

**getline(cin, sgeneral\_secretary);**

**cout << "Input the Budget: ";**

**cin >> ibudget;**

**cout << "Input the total number of events: ";**

**cin >> icnt;**

**for(int i = 0; i < icnt; i++)**

**{**

**cout << "Event " << i+1 << ": ";**

**getline(cin >> ws, sevents[i]); // ws is used to remove the cin skip.**

**}**

**}**

**StudentSociety(int budget, string soc\_name, string fac\_head, string pres, string gen\_sec): ibudget(budget), ssociety\_name(soc\_name), sfaculty\_head(fac\_head), spresident(pres), sgeneral\_secretary(gen\_sec)**

**{**

**// StudentSociety(string)**

**// Overloaded constructor**

**}**

**StudentSociety(int bud): ibudget(bud), ssociety\_name("Nawaz Sharif"), sfaculty\_head("Imran Khan")**

**{**

**// Another overloaded constructor to initialize object members.**

**// This constuctor will call a function Poll().**

**string \*name;**

**name = Poll();**

**for(int i = 1; i <=2; i++)**

**cout << "WOOHOO!!!" << \*(name + i) << " Won" << endl;**

**}**

**string \*Poll()**

**{**

**cout << "Type the name of the General Secretary you want to ellect: ";**

**getline(cin >> ws, vote[1]);**

**cout << "Type the name of the President you want to ellect: ";**

**getline(cin >> ws, vote[2]);**

**return vote;**

**}**

**};**

**int main()**

**{**

**StudentSociety s1; // An object to call a default constructor.**

**StudentSociety s2(50000, "ACM Cyber", "Ali", "Jawwad", "Musharraf"); // An object to call a parameterized constructor**

**StudentSociety s3(100000);**

**}**

Task 2

// Go to main function to understand how the code works.

#include <iostream>

#include <string>

using namespace std;

class Employees

{

char emp\_office\_code[4];

float emp\_salary;

string fname, lname, address;

public:

// The employee will input the name of the employee and assign them with a code.

Employees()

{

cout << "\t\t\t Employee recruitment section " << endl;

cout << "Welcome to Stein Gate Limited" << endl;

cout << "First Name: ";

cin >> fname;

cout << "Last Name: ";

cin >> lname;

cout << "Address: ";

getline(cin >> ws, address);

cout << "Input the office code: ";

cin >> emp\_office\_code;

employee\_office:

if (emp\_office\_code[0] == 'm')

{

goto employee\_office;

}

cout << "Employee Salary: ";

employee\_sal\_reset:

cin >> emp\_salary;

if (emp\_salary > 100000)

{

goto employee\_sal\_reset;

}

}

void disp\_employee\_info()

{

cout << "\t\t\t\tStein Gate Limited Employee Information" << endl;

cout << "Employee First Name: " << fname << endl;

cout << "Employee Last Name: " << lname << endl;

cout << "Office Code: " << emp\_office\_code << endl;

cout << "Employee Salary: " << emp\_salary << endl;

}

};

class Manager

{

// After every office has a manager, the manager manages the company financially and also is responsible

// in hiring new employees and training them.

string manager\_fname, manager\_lname;

float manager\_salary;

char manager\_office\_code[4];

Employees \*e1;

Employees \*e2;

public:

Manager()

{

cout << "Welcome to Stein Gates Limited!";

cout << endl << "You have been selected as a manager for our company: " << endl;

cout << "First name: ";

cin >> manager\_fname;

cout << "Last Name: ";

cin >> manager\_lname;

}

void settingSalary(float y)

{

manager\_salary = y;

}

void settingOfficeCode()

{

manager\_code:

cout << "Assign the manager with an office code(Must start with 'm'): ";

cin >> manager\_office\_code;

if (manager\_office\_code[0] != 'm')

{

goto manager\_code;

}

}

// This function calls employee constructor.

void set\_emp\_info\_office1()

{

e1 = new Employees();

e2 = new Employees();

}

void set\_emp\_info\_office2()

{

e1 = new Employees();

e2 = new Employees();

}

void disp\_emp\_info()

{

e1->disp\_employee\_info();

e2->disp\_employee\_info();

}

/\*Manager()

{

}\*/

};

class Office

{

/// <summary>

/// Now, every office needs employees and every office is managed by a manager.

/// </summary>

Manager \*m1;

Manager \*m2;

int x;

public:

// The office will assign salary to the manager.

Office()

{

m1 = new Manager();

b:

cout << "How much salary would you like to assign to the manager: ";

cin >> x;

if (x > 100000)

{

goto b;

}

else

{

m1->settingSalary(x);

}

m1->settingOfficeCode();

}

Office(int bb)

{

m2 = new Manager();

d:

cout << "How much salary would you like to assign to the manager: ";

cin >> x;

if (x > 100000)

{

goto d;

}

else

{

m2->settingSalary(x);

}

m2->settingOfficeCode();

}

//The office will be the one assigning the manager an office code.

void call\_emp\_info()

{

m1->set\_emp\_info\_office1();

}

void disp\_emp\_information()

{

m1->disp\_emp\_info();

}

};

class Company

{

// The company created 2 offices to function and manage their business. Their shared pool of amount is 50 million.

float amount;

Office \*o1;

Office \*o2;

public:

// The company pulls all the strings so every action the office performs is called from the company board members.

Company()

{

amount = 50000000;

}

// Calls the Office Constructor

void set\_office()

{

o1 = new Office(); // Office one is creted.

o2 = new Office(2); // Office 2 is created.

o1->call\_emp\_info(); // Office 1's manager will initialize their employees with attributes.

//o2->call\_emp\_info();

}

void call\_class\_employee()

{

o1->disp\_emp\_information();

}

};

int main()

{

// Stein Gate Limited is established (goto class Company)

Company c1;

int ch;

c1.set\_office(); // company establishes an office.

cout << "Press 1 or 0 to display Employee information or close the program.\n";

cin >> ch;

if (ch == 1)

{

c1.call\_class\_employee();

}

}