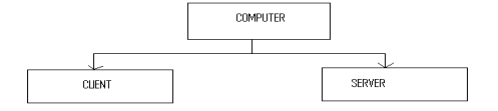
**SAMSON GACHIE MUCHAI**

**ENE211-0042/2020; ASSIGMENT 1**

1. Demonstrate how hierarchical inheritance can be implemented with the following class design.

NB. Include class members (data and functions of your choice). Ensure you implement the classes

using an object in the main() function.



#include <iostream>

#include <string>

using namespace std;

// base class

class Computer {

protected:

string name;

int memory;

public:

void setName(string n) {

name = n;

}

void setMemory(int m) {

memory = m;

}

void printSpecs() {

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

cout << "Memory: " << memory << " GB" << endl;

}

};

// derived class 1

class Server : public Computer {

private:

int speed;

public:

void setSpeed(char c) {

speed = c;

}

void printSpecs() {

Computer::printSpecs();

cout << "Speed: " << speed << endl;

}

};

// derived class 2

class Client : public Computer {

private:

int screenSize;

public:

void setScreenSize(int s) {

screenSize = s;

}

void printSpecs() {

Computer::printSpecs();

cout << "Screen Size: " << screenSize << " inches" << endl;

}

};

int main() {

Server myServer;

myServer.setName("Server computer");

myServer.setMemory(250);

myServer.setSpeed(543);

cout << "Server Specs:" << endl;

myServer.printSpecs();

Client myClient;

myClient.setName("Client Computer");

myClient.setMemory(500);

myClient.setScreenSize(20);

cout << "Client Specs:" << endl;

myClient.printSpecs();

return 0;

}

1. i)Define a class called person with the members age, firstname and lastname and two member methods getValues and setValues as well as a constructor to initialise all the class data members

#include <iostream>

class Person {

private:

int age;

char firstname;

char lastname;

public:

Person(int age, char firstname, char lastname) {

this->age = age;

this->firstname = firstname;

this->lastname = lastname;

}

int getAge() {

return age;

}

void setAge(int age) {

this->age = age;

}

char getFirstname() {

return firstname;

}

void setFirstname(char firstname) {

this->firstname = firstname;

}

char getLastname() {

return lastname;

}

void setLastname(char lastname) {

this->lastname = lastname;

}

};

(ii) Define a class Student that inherits the Person class but adds member variables institution, year (year of study) and registration number.

#include <iostream>

using namespace std;

class Person {

public:

std::string name;

int age;

};

class Student : public Person {

public:

std::string institution;

int year;

std::string reg\_number;

};

1. A university has students taking ICT, Law and Business courses. In the Faculty of ICT, there are Certificate, Diploma and a Degree courses (BSC-IT). BSC-IT has three stages: Stage 1, Stage 2 and Stage 3. (i) Briefly explain how inheritance can assist in the capture and processing of student details in the above scenario. (ii) Develop a program to implement the design in (i)

#include <iostream>

#include <string>

using namespace std;

// base class

class Student {

public:

int studentID;

string name;

string email;

string institution;

void displayDetails() {

cout << "Student ID: " << studentID << endl;

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

cout << "Email: " << email << endl;

cout << "Institution: " << institution<< endl;

}

};

// derived class for ICT students

class ICTStudent : public Student {

public:

string courseType;

int level;

string units;

void displayDetails() {

Student::displayDetails();

cout << "Course Type: " << courseType << endl;

cout << "Level: " << level << endl;

cout << "Units: " << units << endl;

}

void calculateGrades() {

}

};

// main function

int main() {

// create an ICT student object

ICTStudent s;

s.studentID =2110042;

s.name = "Samson";

s.email = "samson@gmail.com";

s.courseType = "BSC-IT";

s.level = 1;

s.units = "Programming, Web design, Com skills";

// display the student details and calculate their grades

s.displayDetails();

s.calculateGrades();

return 0;

}