**COSC 1437 Programming Fundamentals II**

**By KariAnn Harjo**

15 Inheritance and Polymorphism

Exercise 1:

Assume the existence of a Phone class.

Define a derived class, CameraPhone that contains two member variables:

* An int named imageSize that contains the number of megabytes that a picture uses on the phone
* An int named memorySize that contains the number of megabytes in the camera's memory

Write the following member functions in the CameraPhone class:

* A constructor that accepts two int arguments. The first argument should be assigned to the imageSize member, and the second argument should be assigned to the memorySize argument.
* A member function named numPictures that returns (as an int) the number of pictures the camera's memory can hold.

#include <iostream>

#include <string>

using namespace std;

class Phone {

protected:

string model;

public:

Phone(const string& phoneModel) : model(phoneModel) {}

string getModel() const {

return model;

}

};

class CameraPhone : public Phone {

private:

int imageSize;

int memorySize;

public:

CameraPhone(const string& phoneModel, int imageS, int memoryS)

: Phone(phoneModel), imageSize(imageS), memorySize(memoryS) {}

int numPictures() const {

return memorySize / imageSize;

}

};

int main() {

CameraPhone phone("Sample Model", 1024, 4096);

cout << "Model: " << phone.getModel() << endl;

cout << "Number of Pictures: " << phone.numPictures() << endl;

return 0;

}

Exercise 2:

In this exercise you will write the code for two classes.

**Person Class**

Write a class named Person with the following member variables:

* lastName, a string
* firstName, a string
* email, a string
* phone, a string

Write a constructor that accepts no arguments and initializes all the member variables to an empty string.

Write a second constructor that accepts arguments that will be used to initialize the member variables. The arguments must be accepted in this order: lastName, firstName, email, and phone.  
Write mutator functions named setLastName, setFirstName, setEmail, and setPhone.  
Write accessor functions named getLastName, getFirstName, getEmail, and getPhone.

#include <iostream>

#include <string>

using namespace std;

class Person {

protected:

string lastName;

string firstName;

string email;

string phone;

public:

Person() : lastName(""), firstName(""), email(""), phone("") {}

Person(const string& lName, const string& fName, const string& emailAddr, const string& phoneNumber)

: lastName(lName), firstName(fName), email(emailAddr), phone(phoneNumber) {}

void setLastName(const string& lName) {

lastName = lName;

}

void setFirstName(const string& fName) {

firstName = fName;

}

void setEmail(const string& emailAddr) {

email = emailAddr;

}

void setPhone(const string& phoneNumber) {

phone = phoneNumber;

}

string getLastName() const {

return lastName;

}

string getFirstName() const {

return firstName;

}

string getEmail() const {

return email;

}

string getPhone() const {

return phone;

}

};

int main() {

Person person("Doe", "John", "johndoe@example.com", "1234567890");

cout << "Last Name: " << person.getLastName() << endl;

cout << "First Name: " << person.getFirstName() << endl;

cout << "Email: " << person.getEmail() << endl;

cout << "Phone: " << person.getPhone() << endl;

return 0;

}

**Customer Class**

Next, write a class named Customer, which is derived from the Person class. The Customer class should inherit the Person class's member variables and member functions. In addition, the Customer class should have the following member variables of its own:

* customerNumber, an int
* emailList, a bool

The customerNumber member variable will be used to hold a unique integer for each customer. The emailList member variable will be set to true if the customer wishes to be on an email list, or false if the customer does not wish to be on an email list.

Write a constructor that accepts no arguments and initializes customerNumber to 0 and emailList to false. This constructor should also call the base class's constructor that accepts no arguments.  
Write a second constructor that accepts arguments that will be used to initialize the member variables. The arguments must be accepted in this order: lastName, firstName, email, phone, customerNumber, and emailList. This constructor should call the base class constructor to initialize the inherited member variables.  
Write mutator functions named setCustomerNumber and setEmailList.  
Write accessor functions named getCustomerNumber and getEmailList.  
For this exercise, do not write a complete program. Write the classes and their member functions only.

#include <string>

using namespace std;

class Person {

protected:

string lastName;

string firstName;

string email;

string phone;

public:

Person();

Person(const string& lName, const string& fName, const string& emailAddr, const string& phoneNumber);

void setLastName(const string& lName);

void setFirstName(const string& fName);

void setEmail(const string& emailAddr);

void setPhone(const string& phoneNumber);

string getLastName() const;

string getFirstName() const;

string getEmail() const;

string getPhone() const;

};

class Customer : public Person {

private:

int customerNumber;

bool emailList;

public:

Customer();

Customer(const string& lName, const string& fName, const string& emailAddr, const string& phoneNumber, int custNumber, bool emailListStatus);

void setCustomerNumber(int custNumber);

void setEmailList(bool emailListStatus);

int getCustomerNumber() const;

bool getEmailList() const;

};