Bahria University, Islamabad

Department of Software Engineering

Object Oriented Programming

(Spring-2024)

Teacher: Engr. M Waleed Khan

Student : Muhammad Omer Jawaid

Enrollment : 01-131232-063

Lab Journal: 1

Date: 15/02/24

Task No:	Task Wise Marks		Documentation Marks		Total Marks (20)
	Assigned	Obtained	Assigned	Obtained	
1	3		5		
2	3				
3	3				
4	3				
5	3				

Comments:

Signature

LAB

Lab No: 1 – Oop Basics with C++(Classes

Introduction

This lab session explores the basics of 00P, a programming paradigm that uses objects and classes for more efficient and organized code. By the end of this session, students will understand how to create classes, instantiate objects, and enable interactions between different objects, embodying real-world relationships in software design.

Tools Used

VS code

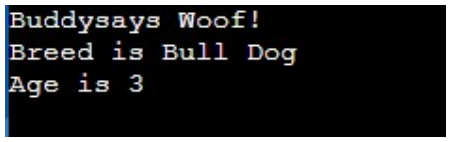
Task 1: Expand the Dogs Class

Code

```
#include <iostream>
using namespace std;

class Dog
{
public:
    string name;
    string breed;
    int age;
    void bark()
    {
        cout << name << "says Woof!" << endl;
    }
    void details()
    {
        cout << "Breed is " << breed << endl
             << "Age is " << age << endl;
    }
};

class Person
{
public:
    string name;
    Dog pet;

    void introducePet()
    {
        cout << "My dog's name is " << pet.name << endl;
        pet.bark();
        pet.details();
    }
};

int main()
```

```
{
    Person person1;
    person1.name = "Alice";
    person1.pet.name = "Buddy";
    person1.pet.breed = "Bull Dog";
    person1.pet.age = 3;
    person1.introducePet();
    return 0;
}
```

Screenshot



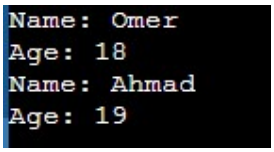
Task 2: Create a House Class

Code

```
#include <iostream>
using namespace std;

class Person
{
public:
    string name;
    int age;
    void persondetails()
    {
        cout << "Name: " << name << endl
             << "Age: " << age << endl;
    }
};

class House
{
public:
    Person person1;
    Person person2;
};

int main()
{
    House house1;
    house1.person1.name = "Omer";
    house1.person1.age = 18;
    house1.person2.name = "Ahmad";
    house1.person2.age = 19;
    house1.person1.persondetails();
```

```
    house1.person2.persondetails();
    return 0;
}
```

Screenshot



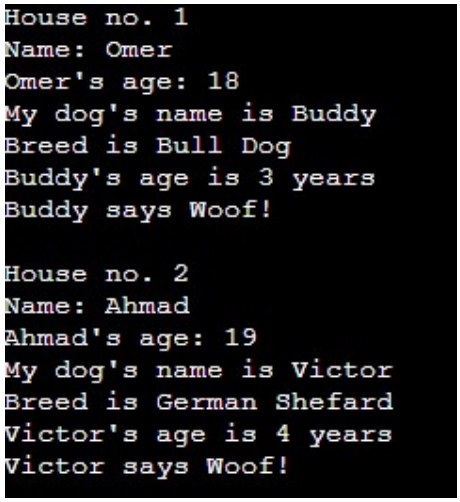
Task 3: Interaction Between Person, Dog and House

Code

```
#include<iostream>
using namespace std;

class Dog {
public:
    string name;
    string breed;
    int age;
    void bark()
    {
        cout << name << " says Woof!" << endl;
    }
    void details()
    {   cout << "Breed is " << breed << endl
            << name<<"'s age is " << age <<" years"<< endl;
    }
};

class Person {
public:
    string name;
    int age;
    Dog pet;
    void persondetails()
    {
        cout << "Name: " << name << endl
            << name<<"'s age: " << age << endl;
        introducePet();
    }

    void introducePet() {
        cout << "My dog's name is " << pet.name << endl;
        pet.details();
        pet.bark();
    }
```

```
};

class House {
public:
    int housenumber;
    Person person1;
    void Details()
    {
        cout << "House no. " << housenumber<<endl;
        person1.persondetails();
    }
};


int main()
{
    House house1;
    House house2;
    house1.housenumber = 1;
    house1.person1.name = "Omer";
    house1.person1.age = 18;
    house1.person1.pet.name = "Buddy";
    house1.person1.pet.breed = "Bull Dog";
    house1.person1.pet.age = 3;
    house1.Details();
    cout << endl;
    house2.housenumber = 2;
    house2.person1.name = "Ahmad";
    house2.person1.age = 19;
    house2.person1.pet.name = "Victor";
    house2.person1.pet.breed = "German Shefard";
    house2.person1.pet.age = 4;
    house2.Details();
    return 0;
}
```

Screenshot



```
House no. 1
Name: Omer
Omer's age: 18
My dog's name is Buddy
Breed is Bull Dog
Buddy's age is 3 years
Buddy says Woof!

House no. 2
Name: Ahmad
Ahmad's age: 19
My dog's name is Victor
Breed is German Shefard
Victor's age is 4 years
Victor says Woof!
```

Task 4: Enhance the Person's Class

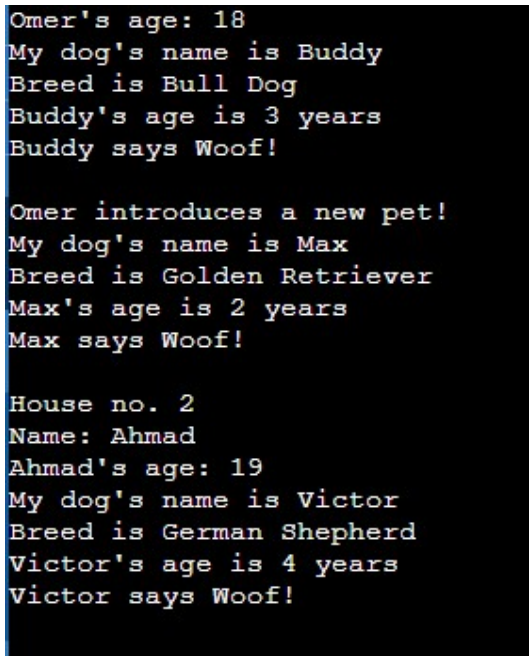
Code

```
#include <iostream>
using namespace std;

class Dog
{
public:
    string name;
    string breed;
    int age;
    void bark()
    {
        cout << name << " says Woof!" << endl;
    }
    void details()
    {
        cout << "Breed is " << breed << endl
             << name << "'s age is " << age << " years" << endl;
    }
};

class Person
{
public:
    string name;
    int age;
    Dog pet;
    void persondetails()
    {
        cout << "Name: " << name << endl
             << name << "'s age: " << age << endl;
```

```
        introducePet();
    }

    void introducePet()
    {
        cout << "My dog's name is " << pet.name << endl;
        pet.details();
        pet.bark();
    }

    void changePet(string newName, string newBreed, int newAge)
    {
        cout << name << " introduces a new pet!" << endl;
        pet.name = newName;
        pet.breed = newBreed;
        pet.age = newAge;
        introducePet();
    }
};

class House
{
public:
    int housenumber;
    Person person1;
    void Details()
    {
        cout << "House no. " << housenumber << endl;
        person1.persondetails();
    }
};

int main()
{
    House house1;
    House house2;
    house1.housenumber = 1;
    house1.person1.name = "Omer";
    house1.person1.age = 18;
    house1.person1.pet.name = "Buddy";
    house1.person1.pet.breed = "Bull Dog";
    house1.person1.pet.age = 3;
    house1.Details();
    cout << endl;
    house1.person1.changePet("Max", "Golden Retriever", 2);
    cout << endl;

    house2.housenumber = 2;
```

```
    house2.person1.name = "Ahmad";
    house2.person1.age = 19;
    house2.person1.pet.name = "Victor";
    house2.person1.pet.breed = "German Shepherd";
    house2.person1.pet.age = 4;
    house2.Details();
    return 0;
}
```

Screenshot



```
Omer's age: 18
My dog's name is Buddy
Breed is Bull Dog
Buddy's age is 3 years
Buddy says Woof!

Omer introduces a new pet!
My dog's name is Max
Breed is Golden Retriever
Max's age is 2 years
Max says Woof!

House no. 2
Name: Ahmad
Ahmad's age: 19
My dog's name is Victor
Breed is German Shepherd
Victor's age is 4 years
Victor says Woof!
```

Task 5: Implement a Neighborhood Class

Code

```
#include <iostream>
using namespace std;

class Dog
{
public:
    string name;
    string breed;
    int age;
    void bark()
    {
        cout << name << " says Woof!" << endl;
    }
    void details()
    {
```

```
        cout << "Breed is " << breed << endl
             << name << "'s age is " << age << " years" << endl;
    }
};

class Person
{
public:
    string name;
    int age;
    Dog pet;
    Person *next = nullptr;
    void persondetails()
    {
        cout << "Name: " << name << endl
             << name << "'s age: " << age << endl;
        introducePet();
    }

    void introducePet()
    {
        cout << "My dog's name is " << pet.name << endl;
        pet.details();
        pet.bark();
    }

    void changePet(string newName, string newBreed, int newAge)
    {
        cout << name << " introduces a new pet!" << endl;
        pet.name = newName;
        pet.breed = newBreed;
        pet.age = newAge;
        introducePet();
    }
};

class House
{
public:
    int number_people;
    Person *person1;
    ~House()
    {
        delete person1;
    }
    void Details(int housenumber)
    {
        cout << "House no. " << housenumber << endl;
```

```
        Person *currentPerson = person1;
        while (currentPerson != nullptr)
        {
            currentPerson->persondetails();
            currentPerson = currentPerson->next;
        }
    }

    void addPerson()
    {
        cout << "How many people live in this house: ";
        cin >> number_people;
        cin.ignore();

        Person *currentPerson = nullptr;
        for (int i = 1; i <= number_people; i++)
        {
            char petchoice;
            Person *newPerson = new Person;
            cout << "Enter Person " << i << " Name: ";
            getline(cin, newPerson->name);
            cout << "Enter Person " << i << " Age: ";
            cin >> newPerson->age;
            cin.ignore();
            cout << "Does " << newPerson->name << " have a pet? (Y/N): ";
            cin >> petchoice;
            cin.ignore();
            if (petchoice == 'Y' || petchoice == 'y')
            {
                cout << "Enter their pet's name: ";
                getline(cin, newPerson->pet.name);
                cout << "Enter their pet's age: ";
                cin >> newPerson->pet.age;
                cin.ignore();
                cout << "Enter " << newPerson->pet.name << "'s breed: ";
                getline(cin, newPerson->pet.breed);
            }
            if (currentPerson == nullptr)
            {
                person1 = newPerson;
                currentPerson = newPerson;
            }
            else
            {
                currentPerson->next = newPerson;
                currentPerson = newPerson;
            }
```

```
        }
    }
};

class Neighborhood
{
public:
    int housenumber;
    House houseobj;
    void addHouses()
    {
        cout << "What is the house number to add: ";
        cin >> housenumber;
        houseobj.addPerson();
        houseobj.Details(housenumber);
    }
};

int main()
{
    string choice;
    Neighborhood neighborhood1;
    cout << "Do you want to add a house to the neighborhood (Y/N)? ";
    getline(cin, choice);
    if (choice == "Y" || choice == "y")
    {
        neighborhood1.addHouses();
    }
    return 0;
}
```

Screenshot

```
Do you want to add a house to the neighborhood (Y/N)? Y
What is the house number to add: 434
How many people live in this house: 2
Enter Person 1 Name: Omer
Enter Person 1 Age: 18
Does Omer have a pet? (Y/N): n
Enter Person 2 Name: Ahmad
Enter Person 2 Age: 17
Does Ahmad have a pet? (Y/N): y
Enter their pet's name: Patty
Enter their pet's age: 3
Enter Patty's breed: Husky
House no. 434
Name: Omer
Omer's age: 18
My dog's name is
Breed is
's age is 0 years
 says Woof!
Name: Ahmad
Ahmad's age: 17
My dog's name is Patty
Breed is Husky
Patty's age is 3 years
Patty says Woof!
```

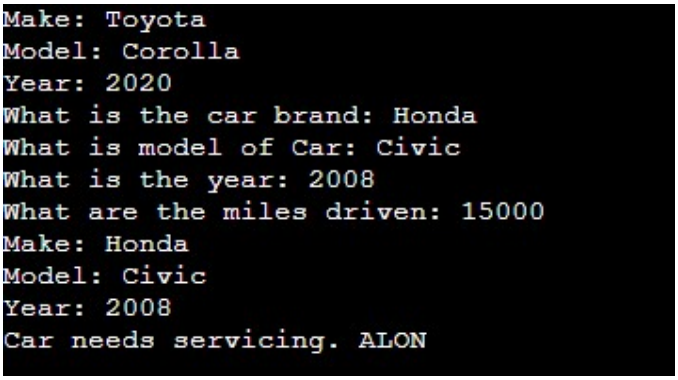
Task 6: Partially Implemented Car class

Code

```cpp
#include <iostream>
using namespace std;

class Vehicle
{
private:
    string make;
    string model;
    int year;

public:
    Vehicle(string mk, string mdl, int yr) : make(mk), model(mdl), year(yr) {}

    void setMake(string mk)
    {
        make = mk;
    }

    void setModel(string mdl)
    {
        model = mdl;
    }
    void setYear(int yr)
    {
        year = yr;
    }
    void displayVehicleInfo()
```

```
    {
        cout << "Make: " << make << "\nModel: " << model << "\nYear: " << year
<< endl;
    }
};
class Car
{
private:
    int miles;

public:
    Car(int mil) : miles(mil) {}
    void service(int mil)
    {
        if (mil >= 10000)
        {
            cout << "Car needs servicing. ALON" << endl;
        }
        else if (mil < 10000 && mil > 0)
            cout << "No service Required" << endl;
        else
            cout << "Enter the correct number of miles!" << endl;
    }
};
int main()
{
    string make, model;
    int year, miles;
    Vehicle myVehicle("Toyota", "Corolla", 2020);
    myVehicle.displayVehicleInfo();
    cout << "What is the car brand: ";
    cin >> make;
    cout << "What is model of Car: ";
    cin >> model;
    cout << "What is the year: ";
    cin >> year;
    cout << "What are the miles driven: ";
    cin  >> miles;
    myVehicle.setMake(make);
    myVehicle.setModel(model);
    myVehicle.setYear(year);
    myVehicle.displayVehicleInfo();
    Car Car1(1000);
    Car1.service(miles);
}
```

Screenshot



Conclusion

This lab introduces 00P basics in C++, guiding students through class and object manipulation, constructors, destructors, and the distinction between classes and structs. These foundational skills are vital for advancing in software development with C++, preparing students for more complex programming challenges. Through hands-on tasks and examples, students will gain practical experience and a deeper understanding of 00P principles.

Git HUB:
https://github.com/OmerJawaid/OOP-LAb-1