OOP (Object Oriented Programming) Lab

LAB REPORT # 9

Semester: 2nd Semester

Section: C

Submitted To:

Mr. Muhammad Husnain

Submitted By:

Name: Abdul Ahad

Roll No: 22-CS-071

Code:

task1.h:

```
#include <iostream>

using std::cout;
using std::endl;
class Base
{
public:
virtual void testFunction ();
};
class Derived : public Base
{
public:
    void testFunction();
};
```

task1.cpp

```
#include "Task1.h"
void Base::testFunction ()
{
cout << "Base class" << endl;
}
void Derived::testFunction ()
{
cout << "Derived class" << endl;
}
```

main1.cpp

```
#include "Task1.h"
int main(void)
{

    Base *ptr = new Base;
    ptr->testFunction(); // prints "Base class"
    delete ptr;
    ptr = new Derived;
    ptr->testFunction(); // prints "Base class" because the base class function is not
virtual
    delete ptr;
    return 0;
}
```

Output:

```
PS F:\MY_PRIVATE_FOLDER\UNI\OOP_LAB\Lab9> g++ task1.cpp main1.cpp -o task1
PS F:\MY_PRIVATE_FOLDER\UNI\OOP_LAB\Lab9> ./task1
Base class
Derived class
```

Code:

mamal.h

```
#pragma once
#include <iostream>
using std::cout;
using std::endl;
class Mammal
{
public:
    Mammal(void);
    ~Mammal(void);

    virtual void Move() const;
    virtual void Speak() const;

protected:
    int itsAge;
};
```

mamal.cpp

```
#include "mamal.h"

Mammal::Mammal(void) : itsAge(1)
{
    cout << "Mammal constructor..." << endl;
}
Mammal::~Mammal(void)
{
    cout << "Mammal destructor..." << endl;
}
void Mammal::Move() const
{
    cout << "Mammal moves a step!" << endl;
}

void Mammal::Speak() const
{
    cout << "What does a mammal speak? Mammilian!" << endl;
}
```

dog.h

```
#pragma once
#include "mamal.h"

class Dog : public Mammal
{
public:
    Dog();
    ~Dog();
```

```
    void Move() const override;
    void Speak() const override;
};
```

dog.cpp

```
#include "dog.h"
#include <iostream>

Dog::Dog()
{
    std::cout << "Dog constructor..." << std::endl;
}

Dog::~Dog()
{
    std::cout << "Dog destructor..." << std::endl;
}

void Dog::Move() const
{
    std::cout << "Dog moves on four legs!" << std::endl;
}

void Dog::Speak() const
{
    std::cout << "Woof! Woof!" << std::endl;
}
```

Main2.cpp

```
#include "mamal.h"
#include "dog.h"

int main()
{
    Mammal *pDog = new Dog;
    pDog->Move();
    pDog->Speak();

    // Dog *pDog2 = new Dog;

    // pDog2->Move();
    // pDog2->Speak();
    return 0;
}
```

Output:

```
PS F:\MY_PRIVATE_FOLDER\UNI\OOP_LAB\Lab9> g++ mamal.cpp dog.cpp main2.cpp -o task2
PS F:\MY_PRIVATE_FOLDER\UNI\OOP_LAB\Lab9> ./task2
Mammal constructor...
Dog constructor...
Dog moves on four legs!
Woof! Woof!
```