

Week 07

Polymorphism

Cảm ơn thầy Trần Duy Quang đã cung cấp template cho môn học



Department of Software Engineering-FIT-VNU-HCMUS

1

Notes

Create a single solution/folder to store your source code in a week.

Then, create a project/sub-folder to store your source code of each assignment.

The source code in an assignment should have at least 3 files:

- A header file (.h): struct definition, function prototypes/definition.
- A source file (.cpp): function implementation.
- Another source file (.cpp): named YourID_Ex01.cpp, main function. Replace 01 by id of an assignment.

Make sure your source code was built correctly. Use many test cases to check your code before submitting to Moodle.

Name of your submission: **StudentID_W05_YY.zip**. YY: number of assignments you have done. YY: 00 – 99.

2

Content

In this lab, we will review the following topics:

1. Apply polymorphism to maintain a list of objects that have the same base class.

3

Assignments

A: YY: 01

H: YY: 03

Draw a class diagram and implement in C++ for each of the following assignments.

Class diagram: https://en.wikipedia.org/wiki/Class_diagram

Use class string and class vector.

3.1. Assignment 1 – Company

Company ABC asks you to build an employee management system.

The system should store the following info for each employee: employ id, full name, hire date, address.

There are 2 types of employees in the company: Office employee and worker.

- Salary of an office employee in a month = number of his/her working days * pay rate per day (300.000 VND/day)
- Salary of a worker in a month = number of items he/she produced in the month * rate pay per item (5.000VND/item)

In this week exercise, you are asked to create 3 classes: Employee, OfficeEmployee and Worker. OfficeEmployee and Worker extends/inherits from Employee.

Then you must define and implement the following methods for each Employee classes.

1. Input the information of an employee
2. Print the information of an employee to console.
3. Compute the salary in a month of an employee
4. Define at least 5 constructors for each of 3 classes.

Then, define another class, Company, store all types of employees in a single array. And write the following methods for class Company:

1. Input a list of employees
2. Print out the info of all employees in the company
3. Print out the total salary the company has to pay in a month
4. Print out all employees whose salaries are the biggest value in the company.

3.2. Assignment 2

```
#include <iostream>
using namespace std;
class A {
private:
    char *m_s;
public:
    A() { m_s = strdup("default"); }
    A(char *s) { m_s = s; }
    virtual void prepare() { cout << "A "; }
    void display() {
        prepare();
        cout << m_s << endl;
    }
};
class B : public A {
public:
    B(char *s) : A(s) { }
    B(const B &b) { }
    void prepare() { cout << "B "; }
};
void foo(A *obj1, A obj2) {
    obj1->display();
    obj2.display();
}
void main() {
    B obj1("text");
    A *obj2 = new B(obj1);
    foo(&obj1, *obj2);
}
```

1. What is printed to the console? Give an brief explanation.
2. Identify the memory issues in the above program and the correct them.
3. Write the input and output operators for class A and class B.

Submit a report for Questions 1 and 2, and provide the full source code for Question 3.

3.3. Assignment 3 – Prince and Princess

A beautiful princess has been imprisoned by a witch in a fortified castle with N layers of gates. To enter the castle and recuse the princess, the prince must pass through all these gates. Each gate is guarded by a gatekeeper, and there are three types of gates.

- **Business Gate:** The gatekeeper here is a merchant. To pass through, the prince must buy an amount of products from the merchant, with the cost caculated as unit price * quantity.
- **Academic gate:** The gatekeeper here is a sage. To pass through, the prince must answer the sage's question. The question has a specified intelligence level and the prince's IQ level must

be equal to or greater than this level to answer correctly. Of course, answering a question does not reduce the prince's IQ.

- **Power gate:** The gatekeeper here is a warrior. To pass through, the prince must defeat the warrior. The warrior has a strength level, and the prince must have a strength level equal to or greater than the warrior's to succeed. After winning the battle, the prince's strength level will be slightly reduced.

Students are asked to apply OOP principles (encapsulation, data hiding, inheritance, polymorphism) to create a class diagram and write a C++ program that performs the following tasks:

1. Input the prince's initial parameters: money, IQ level, strength level, and the information of the castle with n gates.

The program will determine if the prince can rescue the princess. If he can, the program should display the prince's remaining parameters. If not, it should display a list of remaining gates.