Creative Software Programming Final Assignment

Due Dec 18, 2020

P1

Write the code for each `TODO` and describe the output and why.

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

class test {
public:
    static int a;
    static int b;

    template <typename T>
    static void swap(T& a, T& b) {
        cout << "swap refer invoked" << endl;
        // TODO1: swap a and b
    }

    template <typename T>
    static void swap(T* a, T* b) {
        cout << "swap pointer invoked" << endl;
        // TODO2: swap a and b
    }

}

// TODO3:

int main() {
    test::swap(test::a, test::b);
}
```

P2

1. Describe how to find the error in the code below and how to fix it. Also write the result.
2. Change the vehicle's `private` to `protected` and explain if it works, If not, state why.
3. Describe the problem if you declare it as public.

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

class Vehicle {
public:
    Vehicle() : load_(0) { cout << "vehicle created." << endl; }
    ~Vehicle() { cout << "vehicle destroyed." << endl; }

    bool AddLoad(unsigned int weight) {
        load_ += weight;
        cout << weight << " loaded - total load: " << load_ << endl;
        return true;
    }
private:
    unsigned int load_;
};

class Truck : public Vehicle {
public:
    Truck() : Vehicle(), max_load_(10) { cout << "truck create." << endl; }
    ~Truck() { cout << "truck destroyed." << endl; }

    bool AddLoad(unsigned int weight) {
        if (load_ + weight > max_load_) {
            cout << "loading " << weight << "exceeds the max load "
                 << max_load_ << ". " << endl;
            return false;
        }
        return Vehicle::AddLoad(weight);
    }
private:
    unsigned int max_load_;
};

int main() {
    Truck truck;
    Vehicle* veh = &truck;
    if (truck.AddLoad(8) == false) cout << "load failed. " << endl;
    if (truck.AddLoad(3) == false) cout << "load failed. " << endl;
    if (veh->AddLoad(4) == false) cout << "load failed. " << endl;
}
```

P3

Overload `operator+, *, -` with two `std::set`s of union, intersection, and difference. Must use `std::set<T>::iterator` at least once

P4

O/X questions

- ☐ IF the compiler uses the **vtable** mechanism for dynamic binding, an instance of any C++ class has a hidden member variable named **vptr** or similar?
- ☐ The **vptr** of an instance points to the vtable of the instance type, regardless of the types of pointers or references to the instance.
- ☐ Static members can only be accessed by the class name, not by the object instance name?
- ☐ If a function F() is declared as a friend function in the class X, F() can directly access private variables of the class X.

P5

Draw the memory structure. `stack`, `data`, `text`

P6

Explain the result.

```
#include <iostream>

template <class T> void f(T &i) { std::cout << 1; }

template <> void f(const int &i) { std::cout << 2; }

int main() {
  int i = 42;
  f(i);
}
```

P7

Explain the result

```
#include <iostream>

struct X {
  X() { std::cout << "a"; }
  X(const X &x) { std::cout << "b"; }
  const X &operator=(const X &x) {
    std::cout << "c";
    return *this;
  }
};

int main() {
  X x;
  X y(x);
  X z = y;
  z = x;
}
```