C++ Programming

Inheritance Homework 3

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

Homework 1: Thinking

- C++ prevents classes from initializing inherited member variables in the initialization list of a constructor.
  - Why? Think in cases that languages designers considered
- In C++, Friendship is neither inherited nor transitive.
  - Why Friendship functions are not inherited?
- Comparing, the set of objects of a base class with its derived classes, which one is bigger? Think Vehicle vs Honda
- We know that we don’t access private variables, but are they still inherited?
- Think: Making variable protected vs public getter/setter for it if needed?
  - Could subclass corrupts the parent class using protected data?
  - Coupling (dependency) between parent and child?

Homework 2: Package Delivery Service

- Design classes (no main) for a package delivery service (E.g. FedEx):
  - A standard package has a sender address, receiver address, weight in kg and price per kg
    - Total cost is: weight in kg x price per kg
    - Address is: name, string and city
  - A 2-day package is a standard package with an added fixed fee for the total cost
  - A heavy package is a standard package for heavy weights attached with extra weight
    - If weight > 100 kg, then extra fees: (weight - 100) * extra weight in kg
  - Customer can creates several shipments. Each shipment is set of packages of different types
    - Total shipment price = sum of each package price
    - Each shipment has information of card used for payment
    - Customer has several debit and credit card, each has its information

Homework 3: What is the output?

```
4  class A {
5  public:
6      A(string str) { cout<<"Constructor "<<str<<"\n";    }
7      ~A()          { cout<<"~A\n";   }
8  };
9
10 class B {
11     A a1;
12 public:
13     B() : a1(A("Most")) {
14         cout<<"Constructor B"<<"\n";
15     }
16     ~B()          { cout<<"~B\n";   }
17 };
18
19 class C : public B{
20     A a2;
21 public:
22     C() : a2(A("Ali")) {
23         cout<<"Constructor C"<<"\n";
24     }
25     ~C()        {   cout<<"~C\n";   }
26 };
27
28 int main() {
29     C c1;
30     C* c2;
31     return 0;
32 }
```

Homework 4: Guess the problem

```
class A {
protected:
        int px;
        void pf() {
        }
};

class B {
protected:
        int px;
        void pf() {
        }

        int GetSalary() {
                return 100;
        }
};

class C: public A, public B {
public:
        void f() {
                px = 1;
                pf();
        }
        int GetSalary() {
                int parent_salary = GetSalary();
                return 2 * parent_salary + 1;
        }
};
```

- **What** are the problems in this code?
- **Why**?
- **How** may we solve it (syntax)?

Homework 5: Investigate

```
4  class A {
5  public:
6      int x = 1;
7      void print() {      cout << "I am A\n";     }
8      ~A() {      cout << "A Destructor\n";       }
9  };
10
11 class B: public A {
12 public:
13     int y = 2;
14     void print() {      cout << "I am B\n";     }
15     ~B() {      cout << "B Destructor\n";   }
16 };
17
18 class C: public B {
19 public:
20     int z = 3;
21     void print() {      cout << "I am C\n";     }
22     ~C() {      cout << "C Destructor\n";   }
23 };
24
25 void hello(A* a) {
26     a->x = 1;
27     a->print();
28 }
```

```
30 int main() {
31     C* c = new C();
32     A* a_points_c = new C();
33
34     hello(c);
35     hello(a_points_c);
36
37     c->print();
38     a_points_c->print();
39
40     delete c;
41     delete a_points_c;
42
43     return 0;
44 }
```

- What is interesting in the main?
- What is the expected output?
- Which data members (x, y, z) are visible in each object?
    - Same for hello function

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."