Object-Oriented Programming Language

12/13/2018

Homework Assignment No. 10

Due 11:59 pm, Thursday December 20, 2018

Late submission within 24 hours: score*0.9;

Late submission within one week: score*0.8.

The solutions will be posted after one week of the due date.

(Total 100%)

1. **(30%)** We have three classes: `People`, `SmartPeople`, and `VerySmartPeople`. **The `People` class is the parent class, and `SmartPeople` derives `People`, and then `VerySmartPeople` derives `SmartPeople`.**

   In other words, `SmartPeople` is a `People`, and `VerySmartPeople` is a `SmartPeople` and `VerySmartPeople` is also a `People`.

   The following is the definition for the three classes.

```
class People{

public:
    // talk() can be overrided starting in People
    virtual void talk(){
        std::cout<<"People talking.\n";
    }
    void think(){
        std::cout<<"People thinking.\n";
    }
};

class SmartPeople : public People{
public:
    void talk(){
        std::cout<<"SmartPeople talking.\n";
    }
    // think() can be overrided starting in SmartPeople
    virtual void think(){
        std::cout<<"SmartPeople thinking.\n";
    }
};


class VerySmartPeople : public SmartPeople{
public:
    void talk(){
        std::cout<<"VerySmartPeople talking.\n";
    }
    void think(){
        std::cout<<"VerySmartPeople thinking.\n";
    }
};
```

   The main function is as the following:

```
int main(){
    People* p1 = new People();
    People* p2 = new SmartPeople();
    SmartPeople* p3 = new SmartPeople();
    People* p4 = new VerySmartPeople();
    SmartPeople* p5 = new VerySmartPeople();

    p1->talk();
    p1->think();

    p2->talk();
    p2->think();

    p3->talk();
    p3->think();

    p4->talk();
    p4->think();

    p5->talk();
    p5->think();

    delete p1;
    delete p2;
    delete p3;
    delete p4;
    delete p5;

    return 0;
}
```

What are the expected outputs? Please try to figure out the outputs on your own, and then run it on your computer. Did you get it right? What happened? Please document what you have learned in this problem. I hope this helps on how overriding works.

2. **(30%)** Please complete the following program.

```
int main(){
    Rectangle* shape;

    cout << "What do you want to create (Rectangle, Box, or Square)? ";
    char c;
    cin >> c;

    int l = 0, w = 0, h = 0;

    switch(c){
      // fill in your logic here

      default:
          shape = new Rectangle(l,w);
          cout << "Wrong input!" << endl;
    }

    shape->print();
}
```

The following are sample runs:

```
$./a.out
What do you want to create (Rectangle, Box, or Square)? R
Please input the length and width: 5.1 3.2
Rectangle's Length = 5.1 and Width = 3.2
$./a.out
What do you want to create (Rectangle, Box, or Square)? B
Please input the length, width, and height: 5.1 3.2 0.5
Box's Length = 5.1, Width = 3.2 and Height = 0.5
$./a.out
What do you want to create (Rectangle, Box, or Square)? S
Please input the length: 0.8
Square's Length = 0.8
```

3. **(40%)** You will ask the user to input pairs of game names and the corresponding prices. If a game is already in your database, you should inform the user. After the user is done inputting, you will print out all games ordered based on their price. Please complete the following program.

The following is a sample run:

```
Game name: Doom
What is Doom's price? 35
Continue add game (y/n)? y
Game name: Doom
Already have this game!
Game name: JungleBook
What is JungleBook's price? 55
Continue add game (y/n)? y
Game name: LOL
What is LOL's price? 300
Continue add game (y/n)? y
Game name: CounterStreak
What is CounterStreak's price? 100
Continue add game (y/n)? y
Game name: StarWars
What is StarWars's price? 85
Continue add game (y/n)? y
Game name: Rino
What is Rino's price? 120
Continue add game (y/n)? n
Game Doom costs 35
Game JungleBook costs 55
Game StarWars costs 85
Game CounterStreak costs 100
Game Rino costs 120
Game LOL costs 300
```