Lectures and Assignments

C++ Inheritance

Lecture Notes

Lecture 6: C++ Inheritance (PDF)

Lab Exercises

Take a look at this example code:

```
#include <stdio.h>

class Shape {
public:
        virtual ~Shape();
        virtual void draw() = 0;
};

class Circle : public Shape {
public:
        virtual ~Circle();
        virtual void draw();
};

Shape::~Shape() {
        printf("shape destructor\n");
}

// void Shape::draw() {
//      printf("Shape::draw\n");
// }

Circle::~Circle() {
        printf("circle destructor\n");
}
```

```
void Circle::draw() {
        printf("Circle::draw\n");
}

int main() {
        Shape *shape = new Circle;
        shape->draw();
        delete shape;

        return 0;
}
```

Put it in a file named `lab6.cpp` and then compile it like this:

```
$ g++ -Wall lab6.cpp -o lab6
$ ./lab6
Circle::draw
circle destructor
shape destructor
```

Verify your understanding of how the `virtual` keyword and method overriding work by performing a few experiments:

1. Remove the `virtual` keyword from each location individually, recompiling and running each time to see how the output changes. Can you predict what will and will not work?
2. Try making `Shape::draw` non-pure by removing `= 0` from its declaration.
3. Try changing `shape` (in `main()`) from a pointer to a stack-allocated variable.

Assignment 6

rps (CPP)

In the file rps.cpp, implement a class called `Tool`. It should have an `int` field called strength and a `char` field called `type`. You may make them either private or protected. The `Tool` class should also contain the function `void setStrength(int)`, which sets the strength for the `Tool`.

Create 3 more classes called `Rock`, `Paper`, and `Scissors`, which inherit from `Tool`. Each of these classes will need a constructor which will take in an `int` that is used to initialize the `strength` field. The constructor should also initialize the `type` field using `'r'` for `Rock`, `'p'` for `Paper`, and `'s'` for `Scissors`.

These classes will also need a public function `bool fight(Tool)` that compares their strengths in the following way:

- Rock's strength is doubled (temporarily) when fighting scissors, but halved (temporarily) when fighting paper.

- In the same way, paper has the advantage against rock, and scissors against paper.
- The `strength` field shouldn't change in the function, which returns `true` if the original class wins in strength and `false` otherwise.

You may also include any extra auxiliary functions and/or fields in any of these classes. Run the program without changing the main function, and verify that the results are correct.

```
$ g++ -Wall rps.cpp -o rps
$ ./rps
<your test output>
```

Solutions

Solutions are not available for this assignment.