

QUIZ 4: INHERITANCE

1 Problem

It is possible to define square and circle using center of object and only one property - radius for circle and side length for square. This way, it is possible to define rectangle and oval with position and two properties (width and height).

So, in our case, each class has position property. Two classes have one property and two other objects have two properties. Let's try to define these classes using inheritance:

- Class "BasePoint" which will have position properties (x, y)
- Class "Figure1P" which will have one property geometrical figure (p1)
- Class "Square" is based on Figure1P class (has x, y and p1)
- Class "Figure2P" extends Figure1P class with one more property (p2)
- Class "Rectangle" is based on Figure2P (has x, y, p1, p2)

figure.hpp

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

namespace figure{

    //You can add member function in this class
    class BasePoint{
    public:
        BasePoint(int px, int py):x(px),y(py){}
        virtual ~BasePoint(){}
        virtual void info(){
            cout<<"figure: "<<type<<endl;
            cout<<"position x:"<<x<<" y:"<<y<<endl;
        }

    protected:
        int x,y;
    private:
        string type;
    };
    //Fill your code here
}
```

1.1 Sample Test

main.cpp

```
#include "figure.hpp"
#define count 5
using namespace figure;

int main(){
    BasePoint **objs=new BasePoint*[count];
    objs[0]=new BasePoint(1,2);
    objs[1]=new Figure1P(1,2,3);
    objs[2]=new Figure2P(1,2,3,4);
    objs[3]=new Square(1,2,5);
    objs[4]=new Rectangle(1,2,3,4);

    for(int i=0; i<count;i++)
        objs[i]->info();

    for(int i=0; i<count;i++)
        delete objs[i];
    delete [] objs;
}
```

1.2 Output

1.2.1 Format

Your output format should follow the sample output. It has following items :

- "figure": If the object is a geometrical figure(square or circle), show what kind of figure it is. Otherwise, this field is empty.
- "position": The centre point of object.
- "property": Side length for the object.
- "area": If the object is a geometrical figure, you need to print its area.

1.2.2 Sample

```
figure:
position x:1 y:2
figure:
position x:1 y:2
property 1: p1=3
figure:
position x:1 y:2
property 1: p1=3
property 2: p2=4
figure: square
position x:1 y:2
property 1: p1=5
area: 25
figure: rectangle
position x:1 y:2
property 1: p1=3
property 2: p2=4
area: 12
```

1.3 Appendix

A virtual function is a member function that you expect to be redefined in derived classes. When you refer to a derived class object using a pointer or a reference to the base class, you can call a virtual function for that object and execute the derived class's version of the function.

Following code shows difference between using virtual or not:

appendix.cpp

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

class A {
public:
    void print(){ cout << "print A" << endl; };
};

class B: public A {
public:
    virtual void print(){ cout << "print B" << endl; };
};

class C: public B {
public:
    void print(){ cout << "print C" << endl; };
};

int main()d{
    A *a = new C();
    B *b = new C();

    a->print(); // will print 'print A'
    b->print(); // will print 'print C'
}
```

2 How to compile the code at workstation?

You can use this command to compile the code:

```
$ g++ -std=c++14 main.cpp figure.hpp -o main -Wall -Wextra -pedantic -g3
```

3 How to submit the assignment?

Just upload **figure.hpp** to the **e-Campus (E3)** website and do not rename the file or put in into any directory. You will get no credit if you don't follow the rule. **Note: You just can upload the file to e3 ONCE!!**