MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS
CNG242 Programming Language Concepts – Spring 23/24 – Lab 13

PART I: C++ Abstract classes

CPolygon

```
#ifndef CPOLYGON_H
#define CPOLYGON_H
#include <iostream>
#include <string>
using namespace std;
class CPolygon{
protected:
        float width;
        float height;
        static int counter;
public:
        CPolygon(){
                width = 0;
                height = 0;
                counter++;
                cout << "C Polygon Default Constructor" << endl;
        }

        void setup(float width, float height)
        {
                this->width = width;
                this->height = height;
        }

        virtual float area(void) = 0; //pure virtual member function
        virtual string getName(void) = 0; //pure virtual member function

        void onscreen(void){
                cout << this->getName() << ":\t" << this->area() << endl;
        }
        static int getCounter(){
                return counter;
        }
};
int CPolygon::counter = 0;
#endif
```

CRectangle

```
#ifndef CRECTANGLE_H
#define CRECTANGLE_H
#include "CPolygon.h"
class CRectangle : public CPolygon{
public:
        float area(void){
                return width * height;
        }

        string getName(void){
                return "CRectangle";
        }
        void show(void){
                cout << "This is a rectangle" << endl;
        }
};
#endif
```

CTriangle

```
#ifndef CTRIANGLE_H
#define CTRIANGLE_H
#include "CPolygon.h"
class CTriangle : public CPolygon{
public:
        float area(void){
                return (width * height)/2;
        }

        string getName(void){
                return "CTriangle";
        }

        void show(void){
                cout << "This is a triangle" << endl;
        }
};
#endif
```

Main

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

#include "CRectangle.h"
#include "CTriangle.h"

int main(){
        //CPolygon polygon; ERROR
        //Cploygon *p = new CPolygon(); ERROR
        CRectangle rectangle;
        CTriangle triangle;
        triangle.show();

        CPolygon *ptr_polygon1 = &rectangle;
        CPolygon *ptr_polygon2 = &triangle;

        //ptr_polygon2->show() ERROR!
        ptr_polygon1->setup(2, 2);
        ptr_polygon2->setup(2, 2);

        ptr_polygon1->onscreen();
        ptr_polygon2->onscreen();

        cout << "Number of CPolygon created: " << CPolygon::getCounter() << endl;
        system("PAUSE");
        return 0;
}
```

Output

```
C Polygon Default Constructor
C Polygon Default Constructor
This is a triangle
CRectangle:     4
CTriangle:      2
Number of CPolygon created: 2
```

PART II: Prolog

1. **Facts about a hypothetical computer science department:**

```
% lectures(X, Y): person X lectures in course Y
   lectures(turing, 9020).
   lectures(codd, 9311).
   lectures(backus, 9021).
   lectures(ritchie, 9201).
   lectures(minsky, 9414).
   lectures(codd, 9314).

 % studies(X, Y): person X studies in course Y
   studies(fred, 9020).
   studies(jack, 9311).
   studies(jill, 9314).
   studies(jill, 9414).
   studies(henry, 9414).
   studies(henry, 9314).
```

Try:

```
lectures(codd, 9020).
lectures(fred, 9020).
studies (fred, 9020).
lectures(turing, Course).
lectures(codd , Course).
lectures(codd, Course), studies(Student, Course).
```

2. **Let's have an employee table:**

```
employee(193,'Jones','John','173 Elm St.','Hoboken','NJ', 12345,1,'25 Jun 93',25500).
employee(181,'Doe','Betty','11 Spring St.','Paterson','NJ', 12354,3,'12 May 91',28500).
```

What does the following , SQL like relation do?

```
well_paid_emp(First,Last):-
employee(_Num,Last,First,_Addr,_City,_St,_Zip,_Dept,_Date,Sal), Sal > 28000.
```

3. **Analyze the following operations on lists.**

- **list membership**

```
memb(X, [X| Rest ]) .
memb(X, [ _ | Rest ]) :- memb(X, Rest ).
```

- **concatenation**

```
conc([ ] ,L,L).
conc([X|R] , L , [X|RandL]) :- conc(R, L, RandL ).
```

- **second list starts with first list**

```
prefixof([ ] , _ ).
prefixof([X|Rx], [X|Ry]) :- prefixof(Rx, Ry).
```

- **Prefix and suffix using conc**

```
 prefix1(P,L):- conc(P,_,L).
 suffix1(S,L):- conc(_,S,L).
```

4. Sample Calculation Example

```
calculate(0,0).
calculate(X,Y):-X<0, Y is -1.
calculate(X,Y):-X>0, Y is X*X+2*X+1.

calculateV2(0,0).
calculateV2(X,Y):-X<0, Y is -1 ; X>0, Y is X*X+2*X+1.
```

Try:

```
calculate(0,X).
calculate(2,A).
calculate(-2,A).

calculateV2(0,X).
calculateV2(2,A).
calculateV2(-2,A).
```

5. Recursion in Prolog

```
fibo(0,1).
fibo(1,1).
fibo(X,R):-X>0,X1 is X-1, X2 is X-2, fibo(X1,R1), fibo(X2,R2),R is R2+R1.
```

Try:

```
fibo(5,A).
fibo(4,R).
fibo(10,X).
```