Assignment 3

Sunday, April 23, 2023 2:14 PM

COEN 79

Object-Oriented Programming and Advanced Data Structures

Assignment #3

Name: Yen-Jung Lu Date: 04/23/2023

2 points per question, unless noted

1. The sequence's insert member function normally puts a new item before the current item. What does insert do if there is no current item?

I think insert will add the new item at the begining of the sequence, to make it the new current item.

2. (4 pts) Modify the following code to generate the given output. Do not modify the `main` function.

```
1. #include < iostream >
2. using namespace std;
3.
4. class box {
5.
6. public:
7.     // Constructor definition
8.     box(double l = 2.0, double b = 2.0, double h = 2.0) {
9.             length  = l;
10.            breadth = b;
11.            height  = h;
12.            }
13.
14.    double volume() {
15.        return length * breadth * height;  }
16.
17. private:
18.    double length;
19.    double breadth;
20.    double height;
21. };
22.
23. int main(void) {
24.    box Box1(3.3, 1.2, 1.5);   // Declare box1
25.
26.    box Box2(8.5, 6.0, 2.0);   // Declare box2
27.
28.    return 0;
29. }
```

box (double l = 2.0, double b = 2.0, double h = 2.0)
{
length = l;
breadth = b;
height = h;
count ++;
cout << "Number of box objects created so far: " << count << endl;
}

Output:
Number of box objects created so far: 1
Number of box objects created so far: 2

3. (6 pts) In the following code, indicate if the selected lines are legal or illegal:

```
#include <iostream>

class small
{
public:
    small( ) {size = 0;};
    void k() const;
    void h(int i);
    friend void f(small z);

private:
    int size;
};

void small::k() const
{
    small x, y;
    x = y;          // LEGAL/ILLEGAL?
    x.size = y.size; // LEGAL/ILLEGAL?
    size = 3;       // LEGAL/ILLEGAL?
};

void small::h(int i)
{

};

void f(small z)
{
    small x, y;
    x = y; // LEGAL/ILLEGAL?
    x.size = y.size; // LEGAL/ILLEGAL?
    x.size = 3; // LEGAL/ILLEGAL?
    x.h(42); // LEGAL/ILLEGAL?
};

int main() {

    small x, y;
    x = y; // LEGAL/ILLEGAL?
    x.size = y.size; // LEGAL/ILLEGAL?
    x.size = 3; // LEGAL/ILLEGAL?
    x.h(42); // LEGAL/ILLEGAL?

    return 0;
}
```

Legal
Legal
Illegal
Legal
Legal
Legal
Legal
Legal
Legal
Legal

4. (4 pts) We create an array of `fruit` in the `main` function. How can we make sure that for all the items in array `fruit_ptr` the values of `weight` and `color` are equal to 1 and 2, respectively? Please show your solution. Do not modify the main function.

Add public in fruit

class fruit{
public:
fruit: weight(1), color(2){}
};

```
1. class fruit {
2. private:
3.     int weight;
4.     int color;
5. }
6.
7. main() {
8.     fruit * fruit_ptr;
9.     fruit_ptr = new fruit[100];
10. }
```

5. Explain why *heap* variables are essentially global in scope. Please present an example as well.

Heap Variable can be accessed by any function that has a pointer to them,

ex:
```
int *ptr = new int;
*ptr = 42;

delete ptr;
return 0;
```

6. Is it possible to use the keyword "this" inside a friend function? Please explain your answer.

No, it's not possible. "this" is a pointer to the object that called the member function. It is automatically passed as a hidden argument to all non-static member functions. Since friend functions are not members of the class, they don't have the access to "this".

7. (4 pts) Does the following code compile? Does it run? Is there any problem with the code? If yes, how do you fix it?

Yes, I think this code can compile and run.

```
1. #include < iostream >
2. using namespace std;
3.
4. class Computer {
5.     int Id;
6.
7. public:
8.     Computer(int id) {  this -> Id = id;  }
9.     void process() {  cout << "Computer::process()";  }
10. };
11.
12. class Employee {
13.     Computer* c;
14.
15. public:
16.     Employee() {  c = new Computer(123); }
17.     ~Employee() {}
18.     void foo() {
```



```
19.         cout << "Employee::foo()";
20.         c -> process();
21.     }
22. };
23.
24. int main() {
25.     Employee ob;
26.     ob.foo();
27.     return 0;
28. }
```