

实验报告

指导老师：陈家骏 黄书剑

姓名：王晨渊, 学号：181220057

1 概念题

1.1 C++ 中 Lambda 表达式的语法形式是怎样的？Lambda 表达式有什么优点？

用 BNF 可以写为

[< 环境变量使用说明 >] < 形式参数 > < 返回值类型指定 > < 函数体 >

优点在于可以将函数的定义与声明合二为一，同时减少了标识符的使用，为程序的编写带来便利。

1.2 C++ 中类成员访问运算符 -> 可以用于实现”智能指针”的功能。与正常指针相比，”智能指针”有哪些特点？

智能指针可以对堆区上的动态对象进行计数，当所有的指针都不指向某个动态对象时，这个对象可以被自动释放，从而使得程序员无需手动管理内存，大幅改善程序内存泄漏问题。

此外，程序员可以自定义智能指针类进行一些其他操作。

2 编程题

2.1 斐波那契数列

```
1      #include <cassert>
2      class Fib
3      {
4      public:
5          Fib(int index);
6          Fib();
7          int operator()();
8      private:
9          int last;
10         int cur;
11
12     };
13
```

```

14     Fib::Fib(int index)
15     {
16         assert(index >= 1);
17         last = 0;
18         cur = 1;
19         for (int i = 1; i < index; i++)
20         {
21             int temp = last + cur;
22             last = cur;
23             cur = temp;
24         }
25     }
26
27     Fib::Fib()
28     {
29         last = 0;
30         cur = 1;
31     }
32
33     int Fib::operator()()
34     {
35         int temp = last + cur;
36         last = cur;
37         cur = temp;
38         return cur;
39     }

```

2.2 代码填空

```

1     struct Point
2     {
3         int x;
4         int y;
5     };
6
7     bool IsTrue(Point &rstCenter, int iRadius, Point &P1, Point &P2)
8     {
9         auto PointInCircle = [rstCenter, iRadius](Point point)->bool
10        {

```

```

11         return (point.x-rstCenter.x)*( point.x-rstCenter.x)+(point.y-rstCenter.y)
12     };
13     return PointInCircle(P1) != PointInCircle(P2);
14 }

```

2.3 智能指针类

```

1     class B
2     {
3     public:
4         B(A* p);
5         string string_access() const;
6         A* operator ->();
7     private:
8         string last_time;
9         A* p_a;
10    };
11
12    B::B(A* p)
13    {
14        last_time = "No Access!";
15        p_a = p;
16    }
17
18    string B::string_access() const
19    {
20        return last_time;
21    }
22
23    A* B::operator ->()
24    {
25        time_t now_time=time(NULL);
26        last_time =asctime(localtime(&now_time));
27        return p_a;
28    }

```