OOP期末复习试题

110 分

I. 判断题	共 10 小题，共计 10 分
II. 单选题	共 15 小题，共计 30 分
III. 填空题	共 10 小题，共计 32 分
IV. 程序填空题	共 4 小题，共计 38 分

剩余时间: 19:14:41

判断题 单选题 填空题 程序填空题

1-1 Order of initialization in the initial list is the order of their declaration in the list. (1分)

作者: 翁恺
单位: 浙江大学

○ T ◉ F

1-2 An abstract class is a class with at least one pure virtual function. (1分)

作者: 翁恺
单位: 浙江大学

◉ T ○ F

1-3 The operator `::` can not be overloaded. (1分)

作者: 翁恺
单位: 浙江大学

◉ T ○ F

1-4 Dynamic binding is used as default binding method in C++. (1分)

作者: 翁恺
单位: 浙江大学

○ T ◉ F

1-5 One class can have more than one super classes. (1分)

作者: 翁恺
单位: 浙江大学

◉ T ○ F

1-6 The index of an array of pointers to objects starts from 1. (1分)

作者: 翁恺
单位: 浙江大学

○ T ◉ F

1-7 In C++, inheritance allows a derived class to directly access all of the functions and data of its base class. (1分)

作者: 翁恺
单位: 浙江大学

○ T ◉ F

1-8 Destructors can not be overloaded. (1分)

作者: 翁恺
单位: 浙江大学

◉ T ○ F

1-9 Every C++ compiler guarantees that sizeof(int) is less than sizeof(long). (1分)

作者: 翁恺
单位: 浙江大学

○ T ◉ F

1-10 A program is a bunch of objects telling each other how to do by sending messages. (1分)

作者: 翁恺
单位: 浙江大学

○ T ◉ F

保存

判断题

1 2 3 4 5
6 7 8 9 10

单选题

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15

填空题

1 2 3 4 5
6 7 8 9 10

程序填空题

1 2 3 4

OOP期末复习试题

110 分

I. 判断题	共 10 小题，共计 10 分
II. 单选题	共 15 小题，共计 30 分
III. 填空题	共 10 小题，共计 32 分
IV. 程序填空题	共 4 小题，共计 38 分

剩余时间: 19:14:30

判断题 | 单选题 | 填空题 | 程序填空题

2-1 Who can access a private member of a class? (2分)

作者: 翁恺
单位: 浙江大学

- A. Only member functions of that class.
- B. Only member functions of that class and friend functions or member functions of friend classes
- C. Only member functions of that class and derived classes
- D. None of the others

2-2 Resolver `::` is used to: (2分)

作者: 翁恺
单位: 浙江大学

- A. Define a member function outside class declaration
- B. Access a member of a namespace
- C. Access a static member of a class
- D. All of the others

2-3 Given:

作者: 翁恺
单位: 浙江大学

```
class ResId {
public:
                ResId(int Id);
};
```

The expression `ResId res = 5` means: (2分)

- A. A temporary object of class Resld will be created by 5
- B. An object of class Resld will be created by 5
- C. An object of class Resld will be created by a temporary object
- D. The object res will be assiged with the temporary object that was created by 5

2-4 Given:

作者: 翁恺
单位: 浙江大学

```
void f(int i) { cout << "Func1" << endl; }
template<class T>
void f(T t) { cout << "Func2" << endl; }
main() {
        f(2);
}
```

The result is :(2分)

- A. Func1
- B. Func2
- C. *nothing*
- D. undetermined

2-5 Given:

作者: 翁恺
单位: 浙江大学

```
class A {
        A() {};
        virtual f() {};
        int i;
};
```

which statement is NOT true: (2分)

- A. i is private

判断题
1 2 3 4 5
6 7 8 9 10

单选题
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15

填空题
1 2 3 4 5
6 7 8 9 10

程序填空题
1 2 3 4

B. f() is an inline function
C. i is a member of class A
D. sizeof(A) == sizeof(int)

类内定义的函数都是内联函数

2-6 Given:

作者: 翁恺
单位: 浙江大学

```
class A {
        A() {}
        virtual f() = 0;
        int i;
};
```

不能定义抽象类的对象

which statement below is **NOT** true: (2分)

A. i is private
B. Objects of class A can not be created
C. i is a member of class A
D. sizeof(A) == sizeof(int)

2-7 Given:

作者: 翁恺
单位: 浙江大学

```
class X {
        int i;
        virtual void f() {};
};
```

If sizeof(int *) == sizeof(int) == 4, then sizeof(X)==? (2分)

A. 4
B. 6
C. 8
D. Undetermined

类中的非静态成员变量的大小。
虚函数使编译器为类添加一个额外的指针指向虚函数表

2-8 Which one below is copy constructor of class X?(2分)

作者: 翁恺
单位: 浙江大学

A. X()
B. X(const X& x)
C. X(const X& x, int i)
D. X(const X* x)

2-9 Given:

作者: 翁恺
单位: 浙江大学

```
template < class T >
void swap( T& x, T& y ) {
   T temp = x;
   x = y;
   y = temp;
}
int i,j;
float f,m;
```

Which statement is incorrect? (2分)

A. swap(i,j);
B. swap(j,i);
C. swap(f,m)
D. swap(i,f);

2-10 For the code below:

作者: 翁恺
单位: 浙江大学

```
class A {
  static int i;
  //...
};
```

Which statement is NOT true?(2分)

A. All objects of class A reserve a space for i
B. All objects of class A share the space of i
C. i is a member variable of class A
D. i is allocated in global data space

2-11 Which one is the characteristic of abstract class? (2分)

作者: 翁恺
单位: 浙江大学

A. May have virtual functions
B. May have constructors overloaded
C. May have friend function
D. Can not make instance of this class

2-12 In C++ language，function prototype doesn’t identify (2分)

作者: 翁恺
单位: 浙江大学

题目集

题目列表

提交列表

排名

A. The return type of the function.
B. The number of arguments of the function.
C. The functionality of the function
D. The type of arguments of the function.

2-13 In a C++ program, objects communicate each other by (2分)

作者: 翁恺
单位: 浙江大学

A. inheritance
B. encapsulation
C. calling member functions
D. function overloading

2-14 For an arbitrary class，the number of destructors can’t be greater than (2分)

作者: 翁恺
单位: 浙江大学

A. 0
B. 2
C. 1
D. 3

2-15 Suppose a class is defined without any keywords such as public, private and protected，all members default to (2分)

作者: 翁恺
单位: 浙江大学

A. public
B. protected
C. private
D. static

保存

OOP期末复习试题

110 分

I. 判断题	共 10 小题，共计 10 分
II. 单选题	共 15 小题，共计 30 分
III. 填空题	共 10 小题，共计 32 分
IV. 程序填空题	共 4 小题，共计 38 分

剩余时间: 19:14:18

4-1 write the output of the code below.

作者: hulanqing
单位: 浙江大学

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

class INCREMENT
{
public:
   INCREMENT( int v = 0, int i = 1 );
   void addIncrement()
   {
      v += increment;
   }
   void print() const;
   int get() const
   {
          return v;
   }
private:
   int v;
   const int increment;
};

INCREMENT::INCREMENT( int v, int i ) : v( v ),
increment( i )
{
}

void INCREMENT::print() const
{
   cout << v << endl;
}
int main()
{
   INCREMENT value( 1, 2);
   value.print();

   for ( int j = 1; j <= 2; j++ )
   {
      value.addIncrement();
      value.print();
   }
   return 0;
}
```

One for each line:

line 1: 1 (1分) line 2:

3 (1分)

line 3: 5 (1分)

4-2 write the output of the code below.

作者: hulanqing
单位: 浙江大学

判断题

1 2 3 4 5
6 7 8 9 10

单选题

1 2 3 4 5
6 7 8 9 10
11 12 13 14 15

填空题

1 2 3 4 5
6 7 8 9 10

程序填空题

1 2 3 4

```
#include<iostream>
using namespace std;
class TEST
{
    int num;
public:
    TEST( int num=0);
    void increment( ) ;
    ~TEST( );
};
TEST::TEST(int num) : num(num)
{
    cout << num  << endl;
}
void TEST::increment()
{
        num++;
}
TEST::~TEST( )
{
    cout << num  << endl;
}
int main( )
{
        TEST array[2];
        array[0].increment();
        array[1].increment();
        return 0;
}
```

One for each line:

line 1: 0 (1分)

line 2: 0 (1分)

line 3: 1 (1分)

line 4: 1 (1分)

4-3 write the output of the code below.

作者: hulanqing
单位: 浙江大学

```
#include<iostream>
using namespace std;
class Base{
protected:
    int x;
public:
    Base(int b=0): x(b) { }
    virtual void display() const {cout << x << endl;}
};
class Derived: public Base{
    int y;
public:
    Derived(int d=0): y(d) { }
    void display() {cout << x << "," << y << endl;}
};
int main()
{
  Base b(1);
  Derived d(2);
  Base *p = &d;
  b.display();
  d.display();
  p->display();
  return 0;
}
```

One for each line:

line 1: 1 (1分)

line 2: 0,2 (1分)

line 3: 0 (1分)

4-4 write the output of the code below.

作者: hulanqing
单位: 浙江大学

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

enum NOTE { middleC, Csharp, Cflat };
class Instrument {
public:
  virtual void play(NOTE) const = 0;
  virtual char* what() const = 0;
  virtual void adjust(int) = 0;
};

class Wind : public Instrument {
public:
  void play(NOTE) const {
    cout << 1 << endl;
  }
  char* what() const { return "Wind"; }
  void adjust(int) {}
};

class Percussion : public Instrument {
public:
  void play(NOTE) const {
    cout << 2 << endl;
  }
  char* what() const { return "Percussion"; }
  void adjust(int) {}
};

class Stringed : public Instrument {
public:
  void play(NOTE) const {
    cout << 3 << endl;
  }
  char* what() const { return "Stringed"; }
  void adjust(int) {}
};

class Brass : public Wind {
public:
  void play(NOTE) const {
    cout << 11 << endl;
  }
  char* what() const { return "Brass"; }
};

class Woodwind : public Wind {
public:
  void play(NOTE) const {
    cout << 12 << endl;
  }
  char* what() const { return "Woodwind"; }
};

void tune(Instrument& i) {
  i.play(middleC);
}

void f(Instrument& i) { i.adjust(1); }

int main() {
  Wind flute;
  Percussion drum;
  Stringed violin;
  Brass flugelhorn;
  Woodwind recorder;
  tune(flute);
  tune(drum);
  tune(violin);
  tune(flugelhorn);
  tune(recorder);
  f(flugelhorn);
  return 0;
}
```

One for each line:

line 1: 1 (1分)

line 2: 2 (1分)

line 3: 3 (1分)

line 4: 11 (1分)

line 5: 12 (1分)

4-5 write the output of the code below.

作者: hulanqing
单位: 浙江大学

```
#include<iostream>
using namespace std;
class A{
public:
        A& operator=(const A& r)
        {
                cout << 1 << endl;
                return *this;
        }
};
class B{
public:
        B& operator=(const B& r)
        {
                cout << 2 << endl;
                return *this;
        }
};
class C{
private:
        B b;
        A a;
        int c;
};

int main()
{
        C m,n;
        m = n;
        return 0;
}
```

One for each line:

2 (1分)

1 (1分)

4-6 write the output of the code below.

作者: hulanqing
单位: 浙江大学

```
#include <iostream>
using namespace std;
class MYCLASS{
public:
    MYCLASS(int x):val(x){}
    void print() const
    {
        cout << val << endl;
        }
        void print()
    {
        cout << val << endl;
        }
private:
    int val;
};
int main()
{
   MYCLASS ob1(1);
   const MYCLASS ob2(2);
   ob2.print();
   ob1.print();
   return 0;
}
```

One for each line.

line 1: 2 (1分)

line 2: 1 (1分)

4-7 write the output of the code below.

作者: hulanqing
单位: 浙江大学

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

class counter{
private:
        int value;
public:
        counter():value(0) {}
        counter& operator++();
        int operator++(int);
        void reset()
        {
            value = 0;
        }
        operator int() const
        {
            return value;
        }
};

counter& counter::operator++()
{
        if (3 == value)
        value = 0;
        else
            value += 1;
        return *this;
}

int counter::operator++(int)
{
        int t = value;
        if (3 == value)
        value = 0;
        else
            value += 1;
        return t;
}

int main()
{
        counter a;
        while (++a)
             cout << "***\n";
        cout << a << endl;
        while (a++)
             cout << "***\n";
        cout << a << endl;
        return 0;
}
```

One for each line:

***	(1分)
***	(1分)
***	(1分)
0	(1分)
1	(1分)

4-8 write the output of the code below.

作者: hulanqing
单位: 浙江大学

```
#include <iostream>
using namespace std;
int& f(int &i )
{
    i += 10;
    return i ;
}
int main()
{
    int k = 0;
    int& m = f(k);
        cout << k << "#";
    f(m)++;
        cout << k << endl;
        return 0;
}
```

10#21 (2分)

4-9 write the output of the code below.

作者: hulanqing

单位: 浙江大学

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

class Sample{
    friend long fun(Sample s);
public:
        Sample(long a)
        {
            x = a;
        }
private:
        long x;
};
long fun(Sample s)
{
    if (s.x < 2) return 1;
    return s.x * fun(Sample(s.x-1));
}
int main()
{
    int sum = 0;
    for(int i=0;i<6;i++)
        {
           sum += fun(Sample(i));
        }
        cout << sum;
        return 0;
}
```

154 (1分)

4-10 write the output of the code below.

作者: hulanqing
单位: 浙江大学

1.the output at //1 is 1 (1分)

2.the output at //2 is 2 (1分)

3.the output at //3 is 7 (1分)

4.the output at //4 is 0 (1分)

5.the output at //5 is 0 (1分)

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

class Testing
{
private:
        string words;
        int number ;
public:
        Testing(const string & s = "Testing")
        {
                words = s ;
                number = words.length();
                if (words.compare("Testing")==0)
                        cout << 1;
                else if (words.compare("Heap1")==0)
                        cout << 2;
                else
                        cout << 3;
        }
        ~Testing()
        {
                cout << 0;
        }
        void show() const
        {
                cout << number;
        }
};
int main()
{
        Testing *pc1 , *pc2;
        pc1 = new Testing ;          //1
        pc2 = new Testing("Heap1");  //2
        pc1->show();   //3
        delete pc1 ;   //4
        delete pc2 ;   //5
        return 0;
}
```

保存

题目集

题目列表

提交列表

排名

OOP期末复习试题

110 分

I. 判断题	共 10 小题，共计 10 分
II. 单选题	共 15 小题，共计 30 分
III. 填空题	共 10 小题，共计 32 分
IV. 程序填空题	共 4 小题，共计 38 分

剩余时间: 19:14:11

5-1 Run the following program, Enter: 1, the output is: 55 34 21 13 8 5 3 2 1 1

作者: hulanqing
单位: 浙江大学
时间限制: 400 ms
内存限制: 64 MB

判断题
1 2 3 4 5
6 7 8 9 10

单选题
1 2 3 4 5
6 7 8 9 10
11 12 13 14 15

填空题
1 2 3 4 5
6 7 8 9 10

程序填空题
1 2 3 4

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

enum ERROR{UnderFlow,OverFlow};
template<typename T>
class StackTemplate {
        enum { ssize = 100 };
        T stack[ssize];
        int top;
public:
        StackTemplate() : top(0) {}
        void push(const T& i) {
                if (top >= ssize)
trow OverFlow (1分);
                stack[top++] = i;
        }
        T pop() {
                if (top<=0 (1分))
throw UnderFlow;
                return stack[--top] (1分);
        }
        int size() const
        { return top; }
};
int fibonacci(int n);

int main() {
        try (1分) {
                StackTemplate<int> (1分) is;
                for(int i = 0; i < 20; i++)
                        is.push(fibonacci(i));
                for(int k = 0; k < 20; k++)
                        cout << is.pop() << "\t";
        }
        catch( ERROR e ) {
                switch(e (1分))
                {
                case OverFlow:
                        exit;
                case UnderFlow:
                        exit;
                }
        }
        catch(...)
        {
                exit;
        }
        return 0;
}

int fibonacci(int n)
{
        const (1分) int sz = 100;
        int i;
        static int f[sz];
        if (n >= sz) throw OverFlow (1分);
           f[0] = f[1] = 1;
        for(i = 0; i < sz; i++)
                if(f[i] == 0) break;
        while(i <= n) {
                f[i] (1分) = f[i-1]
+ f[i-2];
                i++;
        }
        return f[n] (1分);
}
```

5-2 Run the following program, the output is: B::f()

作者: hulanqing
单位: 浙江大学
时间限制: 400 ms
内存限制: 64 MB

题目集

题目列表

提交列表

排名

```
#include <iostream>
using namespace std;
class A{
public:
     virtual (1分){ cout<<"A::f()\n"; }
 };
class B:public A{
 public:
          void f() {cout<<"B::f()\n"; }
 };
int main()
{
   B b;
   A &p =b (1分);
   p. (1分)f();
   return 0;
}
```

5-3

```
#include <iostream>
using namespace std;
class IndexError{};
template <typename T> (2分)
class ARRAY
{
        size_t m_size;
        T *m_ptr;
public:
        ARRAY(size_t size) : m_size(size)
        {
                m_ptr = new T[size];
                memset(m_ptr, 0, size*sizeof(int));
        }
        ~ARRAY()
        {
                delete[] m_ptr;
        }
        T& at(int index);
};

template <typename T>
T& ARRAY<T> (2分)::at(int index)
{
        if(index<0|| index>=m_size (2分))
        {
                trow (2分)
IndexError();
        }
        return m_ptr[index];
}

int main()
{
        ARRAY<int> a(50);
        int i;
        cin >> i;
        try (2分)
    {
                for(int j=0;j<i;j++)
                        a.at(i) = j;
        }
        catch(IndexError e)
        {
                return 0;
        }
        return 0;
}
```

作者: hulanqing
单位: 浙江大学
时间限制: 400 ms
内存限制: 64 MB

5-4 Run the following program, Enter: 1, the output is:
S1 == S2 HfLLO HFLLO

作者: hulanqing
单位: 浙江大学
时间限制: 400 ms
内存限制: 64 MB

```
#include <iostream>
using namespace std;
class ERROR{};
class STRING
{
   char *m_pStr;
   int m_len;
public:
   STRING(char *str=NULL){
      if (str != NULL) {
             m_len = strlen(str);
             m_pStr =                          (1
分);
             strcpy(                           (1分));
      }
      else {
             m_len = 0;
             m_pStr = NULL;
      }
   }
                                  (1分) operator=(char
*str)
   {
                                  (1分) m_pStr ;
      m_len = strlen(str)+1;
      m_pStr = new char[m_len];
      strcpy(                           (1分));
      return                            (1分);
   }

   bool operator==(STRING str)
                                  (1分)
   {
      return (                          (1分)(m_pStr,
str.m_pStr)== 0);
   }
   char operator [] (int i)
                                  (1分)
   {
      if (i<m_len && i>=0) return m_pStr[i];
      throw                             (1分);
   }
   char& operator[](int i)                  
(1分)
   {
            if (i<m_len && i>=0) return m_pStr[i];
            ERROR e;
                                     (1分);
   }
                                  (1分) ostream&
operator<<(ostream& out ,STRING s);
};
ostream& operator<<(ostream& out ,STRING s)
{
   out << s.m_pStr;
   return out;
}

int main()
{
   STRING s1,s2("HeLLO");
   int i;
   cin >> i;
   s1 = s2;
   if (s1 == s2) cout << "S1 == S2\n";
   s1[1] = s1[1] + 1;
   cout << s1 << endl;;

                                  (1分){
       if(s1[i]>='a' && s1[i]<='z') s1[i] =  s1[i]
- 32;
            cout << s1 << endl;
   }
                                  (1分)(   ERROR& e)
   {
      cout << "upperbound overflow";
   }
```

题目集

题目列表

提交列表

排名

```
    return 0;
}
```

保存