

quiz4. Code=1 Digipen login:_____

1. Problem (6 * 1 pts):

Given the following definitions

<pre>class B { public: virtual void f1(); void f2(); };</pre>	<pre>class D : public B { public: void f1(); void f2(); void f3(); };</pre>
---	---

```
B b;
D d;
B *pb1 = &b, *pb2 = &d;
```

corresponding function of which class is called for each of the following statement, choose NC if does not compile.

A) NC B) B::fx() C) D::fx()	1-1. _____ b.f2();
	1-2. _____ pb1->f3();
	1-3. _____ pb1->f2();
	1-4. _____ pb1->f1();
	1-5. _____ pb2->f3();
	1-6. _____ d.f3();

2. **Problem** (2 * 4 pts):

Given these classes

```
class B {  
    public: virtual std::string name() { return "B"; }  
};  
  
class D : public B {  
    public: virtual std::string name() { return "D"; }  
};
```

what is the output of each of the following **main**s? Notice that each **main** uses a different function **foo**.

- A) In foo: D In main: D
- B) In foo: B In main: D
- C) In foo: B In main: B
- D) In foo: D In main: B

2-1. _____

```
B foo(B& b) { std::cout << "In foo: " << b.name(); return b; }  
  
int main() {  
    D d;  
    std::cout << "In main: " << foo(d).name();  
}
```

2-2. _____

```
B foo(B b) { std::cout << "In foo: " << b.name(); return b; }  
  
int main() {  
    D d;  
    std::cout << "In main: " << foo(d).name();  
}
```

3. **Problem** (3 * 3 pts):

What is the output of the program shown below for each of the following values of "???"

```
void foo(int val) {
    int i = 5; double d = 15.5;
    std::cout << "1";          //PRINT STATEMENT
    switch (val) {
        case 1: //throw int
            throw i; break;
        case 2: //throw double
            throw d; break;
        case 3: //throw nothing
            break;
    }
    std::cout << "2";          //PRINT STATEMENT
}

int main() {
    std::cout << "3";          //PRINT STATEMENT
    try {
        foo(???); // ??? is substituted by a number - see below
    }
    catch (int ex) {
        std::cout << "4";      //PRINT STATEMENT
    }
    std::cout << "5";          //PRINT STATEMENT
}
```

A) 3125 B) 31245 C) 3124 D) 3145 E) 312 F) 314 G) 31 H) 315

3-1. ____ if "???" is substituted by 1

3-2. ____ if "???" is substituted by 2

3-3. ____ if "???" is substituted by 3

4. **Problem** (3 * 2 pts):

Choose a declaration for function `foo` that promises

A) <code>void foo();</code>	4-1. _____ to throw nothing
B) <code>void foo() throw();</code>	4-2. _____ to throw int only
C) <code>void foo() throw(...);</code>	4-3. _____ to throw anything
D) <code>void foo() throw(int);</code>	
E) <code>void foo() throw;</code>	

5. **Problem** (5 * 2 pts):

Given the definitions:

```
template <typename T> void foo(T a)      { cout << "1"; }
template <>          void foo(int a)    { cout << "2"; }
```

```
template <typename T> void foo(T* a)     { cout << "3"; }
template <>          void foo(int* a)    { cout << "4"; }
template <>          void foo(double* a) { cout << "5"; }
```

```
void foo(int* a)    { cout << "6"; }
```

```
double d=1.0; int i=7; char ch='a';
```

What is printed for each of the following, choose "does not compile" if code is illegal?

A) 1	5-1. _____ <code>foo(d);</code>
B) 2	5-2. _____ <code>foo(&i);</code>
C) 3	5-3. _____ <code>foo(&d);</code>
D) does not compile	5-4. _____ <code>foo<int>(&i);</code>
E) 5	5-5. _____ <code>foo<double>(d);</code>
F) 4	
G) 6	

6. **Problem** (8 * 1 pts):

Given the three classes defined below, answer whether each of the following statements compiles or not:

<pre>template <typename T1 = int, int T2 = 10> class Bar { public: Bar(int x = 0) { } private: T1 items[T2]; };</pre>	<pre>class A { public: A() { } };</pre>	<pre>class B { public: B(int x) : x_(x) { } operator int(void) { return x_; } private: int x_; };</pre>
---	---	---

A) Does not compile B) Compiles

6-1. _____ `Bar<int, 5> bar1;`

6-2. _____ `Bar bar2(5);`

6-3. _____ `Bar<A> bar4(B(5));`

6-4. _____ `Bar<B, 5> bar5;`

6-5. _____ `Bar<> bar8;`

6-6. _____ `Bar<5> bar9;`

6-7. _____ `Bar<A, 5> bar10;`

6-8. _____ `int size = 8; Bar<int, size> bar11;`

7. **Problem** (4 pts):

When must template *class* have explicit template parameters?

- A) Always
- B) When the template types cannot be inferred
- C) Never, the template types can always be inferred

8. **Problem** (15 * 1 pts):

Let "cont" be an STL container. Answer whether the following lines compile for a given container type. Assume that container has more than 20 elements, and "iter" is an iterator corresponding to the container.

- A) does not compile B) compiles

- 8-1. ____ `int i = cont[10]; //vector<int>`
- 8-2. ____ `int i = cont[10]; //list<int>`
- 8-3. ____ `int i = cont[10]; //set<int>`
- 8-4. ____ `cont.insert(10); //vector<int>`
- 8-5. ____ `cont.insert(10); //list<int>`
- 8-6. ____ `cont.insert(10); //set<int>`
- 8-7. ____ `iter=cont.begin(); iter++; //vector<int>`
- 8-8. ____ `iter=cont.begin(); iter++; //list<int>`
- 8-9. ____ `iter=cont.begin(); iter++; //set<int>`
- 8-10. ____ `iter=cont.begin(); iter+5; //vector<int>`
- 8-11. ____ `iter=cont.begin(); iter+5; //list<int>`
- 8-12. ____ `iter=cont.begin(); iter+5; //set<int>`
- 8-13. ____ `iter=cont.begin(); *iter=5; //vector<int>`
- 8-14. ____ `iter=cont.begin(); *iter=5; //list<int>`
- 8-15. ____ `iter=cont.begin(); *iter=5; //set<int>`

9. **Problem** (5 pts):

Given

```
template <typename T>
void foo(T t) {}
```

the call

```
std::string str("some huge text");
foo(str);
```

passes `str` by value, which may be very expensive. How do you force compiler to pass `str` by reference without modifying function `foo`?

10. **Problem** (6 pts):

```
class B {
public:
    B() {}
    ~B() {}
};

class D : public B {
    int * pi;
public:
    D(int i) : pi( new (i) ) {}
    ~D() { delete pi; }
    //other methods: copy, assign, ctor are defined here
    //do not implement - they are not relevant to the problem
};

int main() {
    B* pd = new D(100);
    delete pd;
}
```

Are there any problems with the above code at compile-time or run-time? Identify the problems and fix the code – **do not modify** main.

11. **Problem** (4 pts):

```
class B {
public:
    B(int _i);
    B& operator=(const B& rhs) { i=rhs.i; }
private:
    int i;
};

class D : public B {
public:
    D& operator=(const D& rhs);
};
```

Implement derived assignment operator – **do not modify class B**. Notice that D cannot access B::i directly, since the latter is private.

12. **Problem** (5 pts):

What is the difference between single element `delete` and array `delete []`?

Calculate memory leak size for the following program (show work):

```
class C {
    int * array;
public:
    C() : array( new int [10] ) {}
    ~C() { delete [] array; }
};

int main() {
    C* array_of_C = new C[10];
    delete array_of_C; // ERROR !!!
}
```

13. **Problem** (10 pts):

```
#include <iostream>
class Vector3 {
public:
    Vector3() : v(new int[3]) {
        for (unsigned i=0;i<3; ++i) v[i]=0;
    }
    //appropriate copy ctor and assignment operator
    //do not implement - they are not relevant to this problem
    ~Vector3() { delete [] v; }
    int& operator[] (const int & index) {
        return v[index];
    }
private:
    int * v;
};
```

Which lines of this main DO NOT COMPILE?

```
int main() {
    { Vector3 v;          int i = v[1]; } //line 1
    { const Vector3 v; int i = v[1]; } //line 2
    { Vector3 v;          v[1] = 5; }    //line 3
    { const Vector3 v; v[1] = 5; }      //line 4
}
```

Which of the 4 lines of main compile?

From the client's point of view - which line(s) of the main SHOULD compile and which SHOULD NOT?

Modify `operator[]` (add new methods if needed) so that `Vector3` works correctly from the client's point of view.

14. **Problem** (10 pts):

```
class C; //forward declaration

class Cpointer {

    public:

};

class C { //do not modify
    int i;
    public:
    C(int _i=0) : i(_i) {}
    Cpointer operator& () { return Cpointer(this); }
    void Do() { std::cout << "i=" << i << std::endl; }
};

int main() {
    Cpointer p (new C(100));
    p->Do();
}
```

Complete class `Cpointer` so that the code compiles. Make sure there are no memory leaks.

15. **Problem** (6 pts):

What is the output of this program?

```
class Foo {
public:
    Foo() try : pi( int [-1] ) // int [-1] throws an exception
    {
        std::cout << "Foo(int)\n";
    }
    catch (...) {
        std::cout << "catch(...)\n";
    }
    ~Foo() {
        std::cout << "~Foo()\n";
    }
private:
    int* pi;
};

int main () {
    try {
        Foo ();
        std::cout << "try {...}\n";
    }
    catch(...) {
        std::cout << "catch(...) in main\n";
    }
}
```

16. **Problem** (10 pts):

The standard `copy` algorithm copies a range of elements from a source range into a destination. However, there is no standard `copy_if` algorithm. A `copy_if` algorithm performs the copy when the specified predicate returns *true*.

Implement a templated `copy_if` algorithm. (Hint: The function takes 4 parameters.)

```
template <
    copy_if(
```

What is your return value?

Which category of iterators is/are required in your implementation?

17. **Problem** (12 pts):

In this question you are required to implement something similar to `std::bind2nd`. For simplicity `main` is written using a "for"-loop, but it may be rewritten with "for_each" and STL containers.

```
class Functor1 {
public:
    int operator()(const int & i, const int & j) const {
        return i+j;
    }
};

class Functor2 {
public:
    int operator()(const int & i, const int & j) const {
        return i*j;
    }
};

template <typename T>
class BindSecArg

};

int main () {
    Functor1 f1;
    for (int i=0; i<10; ++i) std::cout << f1(i,i) << " "; //0 2 4 6 8 10
    std::cout << std::endl;

    Functor2 f2;
    for (int i=0; i<10; ++i) std::cout << f2(i,i) << " "; //0 1 4 9 16 25
    std::cout << std::endl;

    BindSecArg<Functor1> b1(4); //bind second argument of Functor1 to 4
    for (int i=0; i<10; ++i) std::cout << b1(i) << " "; //4 5 6 7 8 9
    std::cout << std::endl;

    BindSecArg<Functor2> b2(4); //bind second argument of Functor2 to 4
    for (int i=0; i<10; ++i) std::cout << b2(i) << " "; //0 4 8 12 16 20
    std::cout << std::endl;
}
```

Extra credit question: your implementation most probably doesn't work (which is OK!) with

```
class Functor3 {  
    public:  
        std::string operator()(const std::string & i, const std::string & j) const {  
            return i+j;  
        }  
};
```

how does STL solve this problem?