quiz2. Code=1 Digipen login:__________________________

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

   There is a serious problem with `Outer` class constructor.

   - What is(are) the problem(s)?
   - Fix it. Do not modify class `Inner`.

```
class Inner {
  public:
    Inner(int _i) : i(_i) {}
  private:
    int i;
};

class Outer {
  public:
    Outer(int _i) { in.i = _i; }
  private:
    Inner in;
};
```

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

   Code below does not compile. Modify class A to fix the problem. You are not allowed to change `foo` and `main`. Provide program output.

```
#include <iostream>
class A {
    private:
        int i;
    public:
        A()  : i(0) { std::cout << "A()\n"; }
        A(int _i)  : i(_i) { std::cout << "A(int)\n"; }
        ˜A() { std::cout << "˜A()\n"; }
};

int foo(A a) { return a; }

int main(void) {
    int res = foo(3);
}
```

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

```
#include <iostream>
class Vector3 {
  public:
    Vector3() : v(new int[3]) { for (unsigned i=0;i<3; ++i) v[i]=0; }
    ~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
}
```

From the client's point of view - **which** line(s) of the main **should** compile?

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

4. **Problem** (6 * 3 pts):

Which of 7 methods below compile? Notice that they are all syntactically sound.

```
class C {
  public:
    C() { data = new int (100); }
    ~C() { delete data; }
    int        GetInt()         const { return *data; }
    int&       GetRefInt()      const { return *data; }
    const int& GetRefConstInt() const { return *data; }

    int        MemGetInt()         const { return member; }
    int&       MemGetRefInt()      const { return member; }
    const int& MemGetRefConstInt() const { return member; }

  private:
    int * data;
    int   member;
};
```

A) fails to compile B) compiles	4-1.______ GetInt() 4-2.______ GetRefInt() 4-3.______ GetRefConstInt() 4-4.______ MemGetInt() 4-5.______ MemGetRefInt() 4-6.______ MemGetRefConstInt()

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

```
class Foo {
  public:
    Foo() : vec(new int[3]) {
      for (unsigned i=0;i<3; ++i) {
        vec[i]=0;
        arr[i]=0;
      }
    }
    int & foo1 ( int pos ) const { return vec[pos]; }
    int & foo2 ( int pos ) const { return arr[pos]; }
    int & foo3 ( int pos )       { return vec[pos]; }
    int & foo4 ( int pos )       { return arr[pos]; }
    const int & bar1 ( int pos ) const { return vec[pos]; }
    const int & bar2 ( int pos ) const { return arr[pos]; }
    const int & bar3 ( int pos )       { return vec[pos]; }
    const int & bar4 ( int pos )       { return arr[pos]; }
    int baz1 ( int pos ) const { return vec[pos]; }
    int baz2 ( int pos ) const { return arr[pos]; }
    int baz3 ( int pos )       { return vec[pos]; }
    int baz4 ( int pos )       { return arr[pos]; }

  private:
    int * vec;
    int   arr[3];
};
```

A) 1,2,3 B) 2,3,4 C) 1,2,4 D) 2,4 E) 3,4 F) 1,2,3,4 G) 1,3,4 H) 1,4 I) 2,3 J) 1,2 K) 1,3	5-1.______ Which of the `foo` methods COMPILE? 5-2.______ Which of the `bar` methods COMPILE? 5-3.______ Which of the `baz` methods COMPILE?

6. **Problem** (5 * 1 pts):

Given the class definition, determine whether the variable can be accessed or not (that is - wheher expression compiles or not).

```
  class C {
      int i;
    public:
      int j;
      static int si;
    private:
      int k;
  };
```

Assume c is an object of type C.

A) does not compile B) compiles	6-1.______ `c.i;` 6-2.______ `c.j;` 6-3.______ `c.k;` 6-4.______ `c.si;` 6-5.______ `C::si;`

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

```
#include <iostream>
class C {
  public:
  C(int i=0) { std::cout << "C(int)\n"; }
  C(const C& rhs) { std::cout << "C(const&)\n"; }
  C& operator=(const C& rhs) {
    std::cout << "operator=(const C&)\n";
    return *this;
  }
  ~C() { std::cout << "~C()\n"; }
};

C foo() {
  C c;
  std::cout << "no RVO\n";
  return c;
}

int main() {
  C c=5;
  c=foo();
}
```

what is the output of this program? Assume compiler does not perform return value optimization (RVO) in `foo`.

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

Given 3 overloads of function `f6`

```
void f6(short s,  bool  b)  {std::cout << "1";}
void f6(int   i,  bool  b)  {std::cout << "2";}
void f6(float f,  int   i)  {std::cout << "3";}
assume the following declaration
char c = ’a’; int  i = 1; bool b = true; float f = 1.0f; short s = 1;
```

what is printed?

A) 2 B) 1 C) does not compile (ambiguous) D) 3	8-1.______ `f6(c,b);` 8-2.______ `f6(c,s);` 8-3.______ `f6(i,i);` 8-4.______ `f6(f,b);` 8-5.______ `f6(f,c);`

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

Write function prototype for `min`, so that the code

```
int i = 2, j = 3;
min(i,j)=0;
```

compiles.