Midterm Practice

1. True/False.

a) struct members default to private while class members default to public.

b) Except for default access modifiers, a class is identical to a struct.

c) An enclosing class gets automatic access to private members of a nested class inside it.

d) friend overwrites private.

e) Destructors can't be virtual.

2. What is memory leak?

3. What are the 3 numbers the code below is printing?

```
int x, y;
int *px, *py;

int f () {
int s = *px + *py;
cout << s << endl;
return s;
}

int main() {
   x = y = 2;
   px = &x; py = &y;
   x = y = f();
   cout << f() << endl;
}
```

4. What are the 3 numbers the code below is printing?

```
int x, y;
int *px, *py;

int f () {
static int s = *px + *py;
cout << s << endl;
return s;
}

int main() {
   x = y = 2;
   px = &x; py = &y;
   x = y = f();
   cout << f() << endl;
}
```

5. What are the 6 numbers the following code fragment is printing?

```
int x, y;
int *px, *py;

int f ( int a, int& b ) {
      static int s = *px + *py;
      cout << s << endl;
      x = a + s; y = b + s;
      a = x;
      b = y;
      s = x - y;
      cout << s << endl;
      return s;
}

int main() {
      x = y = 1;
      int a = 2, b = 2;
      px = &x; py = &y;
      x = f(a, b);
      b = f(a, b);
      cout << *px << endl;
      cout << *py << endl;
      return 0;
}
```

6. What are the 6 numbers the following code fragment is printing?

```
int x, y;
int *px, *py;

int f ( int a, int b ) {
      static int s = *px + *py;
      cout << s << endl;
      x = a + s; y = b + s;
      a = x;
      b = y;
      s = x - y;
      cout << s << endl;
      return s;
}
int main() {
      x = y = 1;
      int a = 2, b = 2;
      px = &x; py = &y;
      x = f(a, b);
      b = f(a, b);
      cout << *px << endl;
      cout << *py << endl;
      return 0
}
```

7. Write a code fragment allowing you to add 20 C objects to a std::vector.

```
class C {
  int a, b;
 public:
  C (int iA, int iB) :
    a(iA), b(iB) {}
 };
```

8. What is the following code fragment printing?

```
struct A {
 public:
  A() { cout << "A\n"; }
};
class B : public A {
 public:
  B() { cout << "B\n"; }
};

void main() {
 if ( true ) { B b; }
 A* a = new B;
 delete a;
}
```

9. What is the following code fragment printing?

```
struct A {
 public:
  A() { cout << "A\n"; }
 ~A() { cout << "~A\n"; }
};
class B : public A {
 public:
  B() { cout << "B\n"; }
 ~B() { cout << "~B\n"; }
};

void main() {
 if ( true ) { B b; }
 A* a = new B;
 delete a;
}
```

10. What is the following code fragment printing?

```
struct A {
 public:
  A() { cout << "A\n"; }
 virtual ~A() { cout << "~A\n"; }
};
class B : public A {
 public:
  B() { cout << "B\n"; }
 ~B() { cout << "~B\n"; }
};

void main() {
 if ( true ) { B b; }
 A* a = new B;
 delete a;
}
```

11. What is the following code fragment printing?

```
class A {
 public:
  A() { cout << "A\n"; }
  virtual ~A() { cout << "~A\n"; }
};
class B : public A {
 public:
  B() { cout << "B\n"; }
  virtual ~B() { cout << "~B\n"; }
};
class C : public B {
 public:
  C() { cout << "C\n"; }
  virtual ~C() { cout << "~C\n"; }
};

int  main() {
  A* a = new B;
  if ( true ) { C c; }
  delete a;
  return 0;
}
```

12. What is the following code fragment printing?

```
class A {
 public:
  A() { cout << "A\n"; }
  virtual ~A() { cout << "~A\n"; }
};
class B : public A {
 public:
  B() { cout << "B\n"; }
  ~B() { cout << "~B\n"; }
};
class C : public B {
 public:
  C() { cout << "C\n"; }
  ~C() { cout << "~C\n"; }
};

int  main() {
  A* a = new B;
  if ( true ) { C c; }
  delete a;
  return 0;
}
```

13. What is the following code fragment printing?

```
class A {
 public:
  A() { cout << "A\n"; }
  virtual ~A() { cout << "~A\n"; }
};
class B : public A {
 public:
  B() { cout << "B\n"; }
  virtual ~B() { cout << "~B\n"; }
};
class C : public B {
 public:
  C() { cout << "C\n"; }
  virtual ~C() { cout << "~C\n"; }
};

int  main() {
  A* a = new B;
  if ( true ) { C c; }
  B b;
  delete a;
  return 0;
}
```