University of California at Merced

CSE 165 / ENGR 140 Object Oriented Programming

Midterm - Spring 2021

March 30, 2021

Exam start: 9:00am Exam end: 10:15am

NAME: _____

This is an open notes exam. No computing/electronic devices are allowed. Please turn off/silence cell phones. Answer as many questions as possible. Partial credit will be given where appropriate. Be sure to clearly indicate your final answer for each question. Also, be sure to state any assumptions that you are making in your answers.

The exam consists of 7 pages including this one.

You have 75 minutes in which to complete the exam. Good luck!

Problem	Possible Score	Your Score
1	6	
2	6	
3	6	
4	6	
5	6	
6	6	
7	6	
8	8	
9	8	
10	12	
11	10	
12	8	
13	12	
Total	100	

1. Given a code fragment at the right: (6pts)

```
int x;

void func()
{
    int y = 10;
}

int main()
{
    for (int z =0; z < 10; z++)
    {
        ++x;
        func();
    }
}
```

   a. What is the scope of **x**?

Global

   b. What is the scope of **y**?

inside void func

   c. What is the scope of **z**?

the for loop in main

2. Consider the followung fode fragment:

```
struct Big {
      int a[1000]; double b[1000];
};
Big f1() { … }
Big& f2() { … }
```

What is the difference between f1 and f2? (4pts) Which one is more efficient? Why? (2pts)

f1 and f2 are functions of type Big, while f2 returns the address of whatever is passed into it and f1 returns value, f2 is more effective as addresses are easier to manipulate value of variables.

3. a) Define memory leak. (2pts)

A memory leak in a program is when assigned memory at runtime (dynamically allocated) is not treated after its scope, or the end of the program, which can cause problems for the program, at a later stage.

b) Why is the following code fragment generating memory leak? (2pts) How do you fix it? (2pts)

```
void f() {
      classA *a = new classA[200];
}
```

it is generating a memory leak because, we've assigned a pointer to 200 memory of type classA, and haven't done anything with the allocated memory, the allocated memory stays on the heap even after the scope of the f function, to fix this we need to utilize the delete keyword, to delete the wasted memory, delete a;

4. What are the 6 numbers that the following code fragment is printing? (6 points)

```
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;
}
```

Output:

2

0

0

-2

2

4

5. What are the 6 numbers that the following code fragment is printing? (6 points)

```
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;
}
```

Output:

2

0

0

0

2

2

6. What are the 6 numbers that the following code fragment is printing? (6 points)

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

int f ( int a, int b ) {
      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;
}
```

Output:

2

0

4

0

6

6

7. a) What is the difference between a class and a struct? (2pts)

b) Re-write the following code using a class keyword. (4pts)

```
struct S {
      int x, y, z;
      void doSomething() {…}
   private:
      double a;
      void doSomethingElse() {…}
};
```

8. What is the following code fragment printing? (8 pts)

```
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;
}
```

Output:

A
B
A
B
C
~C
~B
~A
~B
~A

9. What is the following code fragment printing? (8 pts)

```
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;
}
```

Output:

A
B
A
B
C
~C
~B
~A
~B
~A

10. What is the following code fragment printing? (12 pts)

```
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;
}
```

Output:

A
B
C
~C
~B
~A
A
B
~B
~A
~B
~A

11. What is the following code fragment printing? (10 pts)

```
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 = new C; }
  B b;
  delete a;
 return 0;
}
```

Output:

A
B
A
B
C
A
B
~B
~A
~B
~A

12. What is the following code fragment printing? (8 pts)

```
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 = new C; }
  B b;
 return 0;
}
```

Output:

A
B
A
B
C
A
B
~B
~A

13. What is the following code fragment printing? (12 pts)

```
class Object{
   public:
      static int count;
      Object(){
          cout << "Object()"<< endl;
            count++;
      }
      ~Object(){
          cout << "~Object()"<< endl;
         count--;
    }
};

int Object::count = 0;

Object f(Object someObject){
    return someObject;
}
int main(int argc, const char * argv[])
{
    Object myObject;
    cout << Object::count << endl;
    Object another = f(myObject);
    cout << Object::count << endl;
    return 0;
}
```

Output:

Object()
1
~Object()
0
~Object()
~Object()