

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

## CS2124 Exam One

### 2019 Spring

Note that I have omitted any `#includes` or “`using namespace std;`” statements in all questions, in order to save space and to save your time thinking about them. You may assume that all such statements that are needed are present. And you don't have to write them either!!!

Please, read all questions carefully! They may *look* familiar and yet be completely different.

Answering the short-answer questions, in particular, requires that you read and *understand* the programs shown. You need to read them *carefully* if you are going to understand them.

If a question asks you to write a class or a function and provides you with test code, be sure your class / function works with that test code. If the question provides you with sample output, then your answer should match that output.

Questions	Points
1	xtra
2-8	7
9	10
10	17
11	17

Answer questions 1–9 in **this exam book**. For multiple choice questions, circle the correct answer. There should be only one correct answer / question.

Answer questions 10 and 11 in your **blue book**.

Place your name and Net id on every page in this book before the end of the exam.

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

**For multiple choice questions, circle one answer!**

1. **[Extra credit]** Who created Java?

- |               |                      |
|---------------|----------------------|
| a) Gosling    | e) Thompson          |
| b) Hopper     | f) van Rosum         |
| c) Ritchie    | g) Wall              |
| d) Stroustrup | h) None of the above |

2. The expression `p->x` means the same thing as:

- |                        |                      |
|------------------------|----------------------|
| a) <code>*p.x</code>   | c) all of the above  |
| b) <code>(*p).x</code> | d) none of the above |

3. Given a class called `Thing` and the code

```
Thing thingOne;
```

```
Thing thingTwo;
```

What **function call** is the following line equivalent to?

```
thingOne = thingTwo;
```

- a) `Thing& Thing::operator=(const Thing& rhs)`
- b) `operator=(thingTwo, thingOne)`
- c) `thingOne.operator=(thingTwo)`
- d) Either (b) or (c), depending on how the programmer chose to implement the operator.
- e) None of the above because it is using the `Thing` copy constructor.
- f) None of the above

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

4. Given:

```
void foo(int* x) {  
    const int* p = &x; // line A  
    x = 17;             // line B  
    cout << *p << ' '; // line C  
    x = 28;             // line D  
}  
  
int main() {  
    int y = 42;  
    foo(y);  
    cout << y << endl;  
}
```

What is the result of compiling and running the above code? (Circle only one answer)

- |   |                                      |
|---|--------------------------------------|
| a) The program will have a compilation error at line A                      | f) The program will print out: 17 17 |
| b) The program will have a compilation error at line B                      | g) The program will print out: 17 42 |
| c) The program will have a compilation error at line C                      | h) The program will print out: 42 17 |
| d) The program will have a compilation error at line D                      | i) The program will print out: 42 42 |
| e) The program will have a runtime error (or undefined behavior) at line D. | j) The program will print out: 17 28 |
|   | k) The program will print out: 42 28 |
|   | l) All of the above                  |
|   | m) None of the above.                |

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

5. Given:

```
struct CandyStruct {  
    protected:  
        CandyStruct () { cout << "sweet\n"; }  
};  
  
class CandyClass {  
    public:  
        CandyClass () { cout << "sweet\n"; }  
};  
  
int main() {  
    CandyStruct candy1;    // line A  
    CandyClass candy2;    // line B  
}
```

What is the result of building and running the above code?

- a) Runs and prints:  
    sweet  
    sweet
- b) Runs with no output because calling the default constructor
- c) compilation error on line A
- d) compilation error on line B
- e) compilation error on both line A and line B
- f) Other compilation error
- g) Runtime error
- h) None of the above

6. Given a vector of doubles, called **doubleVec**, use a ranged for (also known as the “foreach”), to increment each item in the vector  
(No, do not put this in a function.)

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

7. Given:

```
int* foo() {  
    int x = 10;  
    return &x;  
}
```

```
int square(int x) {  
    return x * x;  
}
```

```
int main(){  
    int* y = foo();  
    int z = square(*y);  
    cout << *y << endl;  
}
```

What is the result of building and running the above program?

- |               |                         |
|---------------|-------------------------|
| a) 10         | d) compilation error    |
| b) an address | e) compiles but crashes |
| c) undefined  | f) none of the above    |

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

8. Given:

```
class WeatherReading {
public:
    WeatherReading (string s) : s(s) {}
    // ... possibly other methods that you don't need to know about

private:
    string s;
    // ... possibly other fields that you don't need to know about
};

class Image {
public:
    Image(string s) : name(s), p(new WeatherReading (s)) {}
    ~Image() { delete p; }
private:
    WeatherReading * p;
    string name;
};
```

Implement an appropriate assignment operator, i.e. a deep copy, for the class Image. Write it below.  
Yes, the class WeatherReading supports copy control.

**Your answer here:**

9. Given:

**struct Thing{ int val; };**

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

- a) Define a variable `dptr` that points to a dynamic array of 66 Memo pointers.  
No the pointers will not be assigned / initialized here.
- 

- b) Fill the array with addresses of 66 Memos that you allocate on the heap. Each Memo will have its `id` field hold a value from 1 to 66. i.e. the first Memo will hold 1, the second 2, ...
- 

- c) Now, modify those values by adding the index of the entry to the *value* that was stored on the heap, e.g. add 17 to `id` field of the Memo pointed to by `dptr[17]`.
- 

- d) Finally, free up all of the space you allocated on the heap.
-

Name: \_\_\_\_\_

Net ID: \_\_\_\_\_

### **Blue book**

#### **Answer questions 10 and 11 in your blue book.**

10. Please write a Complex class that supports:

- a) Construction with a real and imaginary component (both doubles), each with a default value of 0.
- b) Private data members to hold the real and imaginary components.
- c) Overrides of +, ++ (pre and post!), and <<.

11. Please write a Vector class that holds ints. The underlying data structure should be a C-style array of ints. You need to write:

- a) A constructor that takes two int parameters, 'size' and 'val', and creates an initial array of size 'size' plus some extra slots for expansion (so you need another variable like 'total' to track how much extra you have), filled with values 'val'.
- b) A method 'void push\_back(int)' that adds items to the array, tracks the new size, and allocates new memory if needed (if we fill up all of the "extra" spaces we have allocated), and cleans up any memory no longer needed.
- c) Overrides of [] for a right-hand-side situation, override of [] for a left-hand-side situation, and override of an output operator.

**Write your answer in the Blue Book!**