



Week 2 Quiz

TOTAL POINTS 10

1. Recall that every variable in C++ has these four things: a name, a type, a value and a memory location.

1 point

```
1  int *p;  
2  p = new int;  
3  *p = 0;  
4  
5
```

For the code above, which one of the following is NOT true for variable p?

- ☐ The name of the variable is "p"
- ☐ The type of the variable is a pointer to an integer, specifically the type "int *"
- ☐ The value of the variable is 0
- ☐ The memory address of the variable is the value returned by the expression &p

2. Which one of the following is true?

1 point

- ☐ The "new" operator allocates memory on the stack that gets removed from the stack by the "delete" operator.
- ☐ The C++ statement "int i;" allocates memory for one integer on the heap.
- ☐ You should avoid using the memory address 0x0 for pointers whose value is not yet set, because memory location 0x0 is a valid location for the system to allocate to hold the contents of a variable.
- ☐ The address of any memory location in the stack is larger than the address of any memory location in the heap.

3. Suppose we are writing the following function that is intended to return a pointer to a location in memory holding an integer value initialized to zero.

1 point

```
1  int *allocate_an_integer() {  
2      // declare variable i here  
3      *i = 0;  
4      return i;  
5  }  
6  
7
```

How should variable i be declared?

- ☐ int j;
int *i = &j;
- ☐ int *i;
- ☐ int i;
- ☐ int *i = new int;

4. Suppose we have this alternative function that returns a pointer to a memory location to an integer value of zero.

1 point

```
1  int *allocate_an_integer() {  
2      int i = 0;
```

```

2   int i = 0;
3   return &i;
4 }
5
6 int main() {
7     int *j;
8     j = allocate_an_integer();
9     int k = *j;
10    return 0;
11 }
12
13

```

What value is variable k assigned and why?

- ☐ Unknown. Depending on the compiler settings, the compiler may report that a local variable address is being returned, which could be treated as a warning or as a compilation error; Or, if the program is allowed to compile, then at runtime the variable k could be assigned zero, or some other value, or the program may terminate due to a memory fault.
- ☐ Variable k is not assigned a value, because even if the compiler is set to ignore warnings and continue with compilation, the compiled program will still automatically detect that a local variable's address is being used after the function has returned, and exit to the operating system with a non-zero error code.
- ☐ Variable k is certainly assigned the value zero, because the C++ runtime will automatically move the local variable to the heap and return the address of that heap variable instead.
- ☐ Assuming that the program compiles with just a warning and not an error due to the settings, the variable k will not be assigned a value, because the running program will crash the whole operating system.

5. Suppose we declare a variable as "int i;" Which of the following expressions returns the address of the memory location containing the contents of variable i?

1 point

- ☐ &i
- ☐ i->addr
- ☐ i.addr
- ☐ *j

6.

1 point

```

1   int i = 0;
2   int *j = &i;
3
4

```

How many memory allocations are made on the stack and on the heap for the above code? For example, declaring an integer would count as one memory allocation.

- ☐ Zero allocations on the stack and one allocation on the heap.
- ☐ One allocation on the stack and zero allocations on the heap.
- ☐ Zero allocations on the stack and two allocations on the heap.
- ☐ One allocation on the stack and one allocation on the heap.
- ☐ Two allocations on the stack and zero allocations on the heap.

7.

1 point

```

1   int *i = new int;

```

How many memory allocations are made on the stack and on the heap for the above code? For example, allocating space for one integer would count as one memory allocation.

- ☐ Zero allocations on the stack and one allocation on the heap.
- ☐ Two allocations on the stack and zero allocations on the heap.
- ☐ Zero allocations on the stack and two allocations on the heap.
- ☐ One allocation on the stack and one allocation on the heap.
- ☐ One allocation on the stack and zero allocations on the heap.

8. 1 point

```

1  int *i = new int;
2  *i = 0;
3  int &j = *i;
4  j++;
5
6

```

What does the last line of the above code segment do?

- ☐ Increments the address pointed to by variable i by one.
- ☐ Increments the value pointed to by variable i by one.
- ☐ Causes an error.
- ☐ Increments the value of j by one, where the value of j is a local copy stored on the stack of the value of i stored on the heap.

9. 1 point

```

1  int i = 0, j = 1;
2  int *ptr = &i;
3
4  i = 2;
5  *ptr = 3;
6  ptr = &j;
7  j = i;
8  *ptr = 4;
9
10

```

Enter the number of different values stored in the same address that variable i has during the execution of the code above. (Your answer should be a single integer, which is the total number of different values assigned to that address.)

Enter answer here

10. 1 point

```

1  class Pair {
2      public: double a,b;
3  };
4
5  int main() {
6      Pair *p = new Pair;
7      p->a = 0.0;
8      return 0;
9  }
10
11

```

The expression p->a is equivalent to which one of the following?

- ☐ p.a
- ☐ (*p).a
- ☐ p.*a
- ☐ *(p.a)

☐

I, **Jiarong Yang**, understand that submitting work that isn't my own may result in permanent failure of this course or deactivation of my Coursera account.

[Learn more about Coursera's Honor Code](#)



Save

Submit