Week 3 Quiz

TOTAL POINTS 10

1. Which one of these is the only true statement about a class constructor. 1 point

- A class can only have one constructor.
- When declaring a constructor for a class, the name of the constructor must be the name of the class.
- When declaring a constructor for a class, the return type of the constructor must be the type of the class.
- A class must have at least one constructor declared for it.

2. Which of the following examples does NOT call a copy constructor at least once? 1 point

(If you already have advanced knowledge of C++ that makes this seem like a trick question, then we'll also specify this: Assume that compiler optimizations are mostly disabled.)

```
// Function prototype for "intersect":
Cube intersect(Cube &left, Cube &right);
// ...
Cube a(10),b(5);
Cube c;
c = intersect(a,b);

```

```
Cube a,b(10);
a = b;
```

```
Cube b(10);
Cube a = b;
```

```
// Function prototype for "contains":
int contains(Cube outer, Cube inner);
// ...
Cube a(10),b(5);
int a_bounds_b = contains(a,b);

```

3. Recall that a custom assignment operator can be declared such that line 2 of the code below executes a user-defined function to perform the assignment. 1 point

```
Cube a,b(10);
a = b(10);
```

Which one of the following statements regarding the declaration of such a custom assignment operator allowing is true?

- The custom assignment operator function is declared with two arguments: the source and target objects of the assignment.
- The type of the custom assignment operator function should be void.
- The custom assignment operator is a public member function of the class.
- The custom assignment operator is a function declared with the name "operator::assignment".

4. Consider the following class: 1 point

```
class Orange {
  public:
    Orange(double weight);
    ~Orange();
    double getWeight();

  private:
    double weight_;
};

```

Select **all** functions that are present in this class (including any automatic/implicit functions added by the compiler):

- [ ] Default constructor
- [ ] At least one custom, non-default constructor
- [ ] Copy constructor
- [ ] Assignment operator
- [ ] Destructor

5. Consider the following class: 1 point

```
class Blue {
  public:
    double getValue();
    void setValue(double value);

  private:
    double value_;
};

```

Select **all** functions that are present in this class (including any automatic/implicit functions added by the compiler):

- [ ] Default constructor
- [ ] At least one custom, non-default constructor
- [ ] Copy constructor
- [ ] Assignment operator
- [ ] Destructor

6. Consider the following class: 1 point

```
class Animal {
  public:
    Animal();
    Animal(std::string name);
    Animal(std::string name, int age);
    Animal(std::string name, int age, double weight);

    Animal(const Animal & other);

    void setName(std::string name);
    std::string getName();

```

```
13    private:
14      // ...
15  };
16
```

How many **explicit** (non-automatic) **constructors** are present in the class?

- 2
- 3
- 4
- 5
- 6
- 7

7. When you use the **new** operator to create a class object instance in heap memory, the **new** operator makes sure that memory is allocated in the heap for the object, and then it initializes the object instance by automatically calling the class constructor. 1 point

After a class object instance has been created in heap memory with **new**, when is the *destructor* usually called?

- The programmer always needs to call the destructor manually in order to free up memory.
- The destructor is called automatically when the program returns from the function where the **new** operator was used to create the class object instance.
- The destructor is called automatically when the variable goes out of scope.
- The destructor is called automatically when the **delete** operator is used with a pointer to the instance of the class.

8. Consider the following program: 1 point

```
1   double magic(uiuc::Cube cube) {
2     cube.setLength(1);
3     return cube.getVolume();
4   }
5
6   int main() {
7     uiuc::Cube c(10);
8     magic(c);
9     return 0;
10  }
11
```

How many times is the **uiuc::Cube** 's copy constructor invoked?

- Never
- Once
- Twice
- Three times

9. We have looked at examples where the assignment operator returned the value "*this". The variable "this" is available by default in most class member functions. What is the value of this built-in class variable "this"? 1 point

- An alias of the current object.
- A pointer to a heap-memory copy of the current object.
- A reference to the current object.
- A pointer to the current object instance.

10. Consider the code below that includes a class that has a custom constructor and destructor and both utilize a global variable (which has global scope and can be accessed anywhere and initialized before the function main is executed). 1 point

```
int reference_count = 0;

class Track {
public:
    Track() { reference_count++; }
    ~Track() { reference_count--; }
};
```

Which one of the following procedures (void functions) properly ensures the deallocation of all the memory allocated for objects of type Track so the memory can be re-used for something else after the procedure returns?

For the correct answer, the variable reference_count should be zero after all calls to track_stuff() **and** all of the memory should be deallocated properly. This will dependably occur after only one of the following procedures.

```
void track_stuff() {
    Track *t = new Track;
    // ...
    t->~Track();
    return;
}
```

```
void track_stuff() {
    Track t;
    Track *p = new Track;
    // ...
    delete p;
    return;
}
```

```
void track_stuff() {
    Track t;
    Track *p = &t;
    // ...
    delete p;
    return;
}
```

```
void track_stuff() {
    Track t;
    // ...
    delete t;
    return;
}
```

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