

1. (10 points) Give the output for the following program:

```
1 #include <iostream>
2 class string {
3 public:
4     string() { std::cout << "default" << std::endl; }
5     string(const char*) { std::cout << "convert" << std::endl; }
6     string(const string&) { std::cout << "copy" << std::endl; }
7     ~string() { std::cout << "destructor" << std::endl; }
8     string& operator=(const string&) {
9         std::cout << "assign" << std::endl;
10        return *this;
11    }
12 };
13 int main() {
14     string a("cat");
15     string* rope = new string("twisted");
16     delete rope;
17 }
```

2. (10 points) Give the output for the following program:

```
1 #include <iostream>
2 void incrCount(int count) {
3     ++count;
4 }
5
6 void makeSwitch(int & count) {
7     switch ( count ) {
8         case 3: ++count;
9         case 4: ++count;
10        case 5: ++count;
11        case 6: ++count;
12        default: ++count;
13    }
14 }
15 int main() {
16     int count = 2;
17     incrCount(count);
18     std::cout << count << std::endl;
19
20     count = 2;
21     count = (count % 2) ? 2 : 3;
```

```
22     makeSwitch(count);
23     std::cout << count << std::endl;
24 }
```

3. (15 points) Give the output for the following program:

```
1 #include <iostream>
2 class string {
3 public:
4     string() { std::cout << "default" << std::endl; }
5     string(const char*) { std::cout << "convert" << std::endl; }
6     string(const string&) { std::cout << "copy" << std::endl; }
7     ~string() { std::cout << "destructor" << std::endl; }
8     string& operator=(const string&) {
9         std::cout << "assign" << std::endl;
10        return *this;
11    }
12 };
13 int main() {
14     string x("cat"), y = x;
15 }
```

4. (15 points) The following program crashes with a double free error. (1) Explain why, and (2) fix the problem by writing the required member function.

```
1 #include <cstring>
2 #include <iostream>
3 class string {
4 public:
5     string(const char* s) : buf(new char[strlen(s)+1]) { strcpy(buf, s); }
6     ~string() { delete [] buf; }
7     const char* getBuf() const { return buf; }
8 private:
9     char * buf;
10 };
11
12 int main() {
13     string a("cat"), b = a;
14 }
```

5. (10 points) Write an output operator for class `string` in the previous question.

6. (10 points) Class `string` contains a method, lines 10–12, that overloads the output operator. Write code on line #21 to use this output method to display `string` variable `star`

```
1  #include <iostream>
2  #include <cstring>
3  class string {
4  public:
5      string(const char* b) : buf(new char[ strlen(b)+1]) {
6          strcpy(buf, b);
7      }
8      const char* getBuf() const { return buf; }
9      ~string() { delete [] buf; }
10     std::ostream& operator<<(std::ostream& out) {
11         return out << buf;
12     }
13 private:
14     char* buf;
15     string& operator=(const string&);
16     string(const string&);
17 };
18 int main( ) {
19     string star("starlord");
20
21     // -----
22 }
```

7. (20 points) The listing below contains a class, `Random`, with an overloaded parentheses operator that returns a random number. Convert `Random` so that it is a *singleton*, and write function `main` so that `main` uses the singleton to get and print a random number.

```
1 #include <cstdlib> // for rand()
2 #include <iostream>
3
4 class Random {
5 public:
6     Random() {
7         int seed = time(0);
8         srand(seed);
9     }
10    int operator()(int a, int b) {
11        return (rand() % b) + a;
12    }
13 private:
14    Random(const Random&);
15    Random& operator=(const Random&);
16 };
```

8. (10 points) The following questions are taken from the first four items in Effective C++, by Meyer's.

- (a) In Item #3, Meyer's says to use `const` whenever possible. What restrictions does the following use of `const` place on `buffer`:

```
const char* buffer = new char[10];
```

- (b) What restrictions are placed on a `const` member function.

- (c) In the third edition of Meyer's text, Item # 4 is titled: "Make sure that objects are initialized before they are used." However, in the second edition Item #4 was titled: "Prefer initialization to assignment in constructors."

What is the difference between initialization and assignment in constructors? When must we use initialization in constructors?