Selected Exercises

Exercises 1-10 refer to the `String` class, described in the lectures.

1. Next to each expression, indicate which constructor or operator is used.

```
String s1;
String s2 = s1;
String s3;
s3 = s1;
String s4 ("glory days by");
String s5;
s5 = s4;
String s6 = s5;
String s7 = String ("Judy blue eyes");
```

2. Define an array of `Strings` and initialize every other element to "widget." Elements not initialized to "widget" should be initialized by the default constructor.

3. Which constructor is at work in this code slice?

```
String* s = new String;
```

4. If the programmer does not write his or her own copy constructor, is the compiler's copy constructor still at work during call by value?

5. Illustrate with pictures how the copy constructor works during a call by value with a `String` object.

6. Why should you not rely on the compiler to overload the assignment operator for class `String`?

7. Illustrate with pictures how the copy constructor works during returning a value with a `String` object.

8. Why does `String::operator=` check whether a `String` object is being assigned to itself? What problem would arise if this check were absent?

9. Illustrate with pictures a problem that could arise if we did not provide `String::operator=` for class String.

10. Can you imagine a situation in which it would be a good idea to write your own copy constructor but not your own assignment operator?

11. Is the copy constructor at work in return by reference? Explain.

12. The copy constructor for a class `x` has the form `x(const x& b)`. Why is the construction parameter passed by reference? Explain why you can't define a constructor of the form `x (x b)`.

13. True or false:
    a. If a program has passed all inputs you test it with, it has no bugs.
    b. If a program has a bug, it will always show when running the program through the debugger.
    c. If each function in a program is shown to be correct, then the program is correct.

14. In `ddd`, what is the difference between stepping onto a function and stepping over a function?

15. Explain how to inspect the string stored in a `String` object using `ddd`.

16. Find the mistakes in the following code. Not all lines contain mistakes. Each line depends on the lines preceding it. Watch out for un-initialized pointers, NULL pointers, pointers to deleted objects, and confusing pointers with objects.

```
int* p = new int;
p=5;
*p = *p + 5;
Time t1 = new Time (1, 10, 40);
Time t2;
t2->PrintTime();
delete t2;
Time* now = new Time ();
Time* t3 = new Time (2, 0, 0);
cout << t3 + now;
delete *t3;
cout << t3->AddToTime();
String * s3 = new String ("Lin, Lisa”);
cout << s3;
Time* t4 = new Time (1, 25, 0);
delete t4;
```

17. A pointer variable can contain a pointer to a valid object, a pointer to a deleted object, NULL, or a random value. Write code that creates and sets four pointer variables `a`, `b`, `c`, and `d` to show each of these possibilities.

18. What happens when you dereference each of the four pointers that you created in the preceding question? Write a test program if you are not sure.

19. What happens if you forget to delete an object that you dynamically allocated? What happens if you delete it twice?

20. Which of the following assignments are legal in C++?

```
void f(int p[]) {
    int* q;
    const int* r;
    int s[10];
    p = q;
```

```
    P = r;
    p = s;
    q = p;
    q = r;
    q = s;
    r = p;
    r = q;
    r = s;
    s = p;
    s = q;
    s = r;
}
```

21. Given the definitions:

```
double values[] = { 2, 3, 5, 7, 11, 13 };
double* p = values + 3;
```

explain the meanings of the following expressions:

a. `values[1]`
b. `values + 1`
c. `*(values + 1)`
d. `p[i]`
e. `p + 1`
f. `p - values`

22. What is the difference between the following three variable definitions:

a. `char * p = NULL:`
b. `char * q = “”;`
c. `char r[] = {‘\0’};`

23. What is the difference between initialization:

```
String s ("abc");
String u = t;
```

and assignment.

```
String s ("abc");
String u ("xyz");
u = s;
```

24. How many times is the copy constructor invoked when the following code segment executes?

```
int main() {
   String  one = "Fred";
   String  two = "Alice";
   String  three = one;
   String  four;
   four = three;
   three = two;
}
```

25. Consider the following class definition:

```
class Golfer {
    private:
        char * fullname;
        int games;
        int * scores;
    public:
        Golfer ();
        Golfer (const char * name);
        Golfer (const char * name, int g);
        Golfer (const Golfer & g);
         ~Golfer ();
};
```

What class methods would be invoked by each of the following statements?

```
Golfer nancy;
Golfer lulu ("little lulu");
Golfer roy ("Roy Hobbs", 12);
Golfer *par = new Golfer ();
Golfer next = lulu;
*par = lulu;
```