

CPRE 416: Software Evolution and Maintenance
Assigned: September 30, 2005
Due: October 7, 2005 at the beginning of the class
Name: Fill in on your answer sheet

Homework 5 (35 points)

You are given two unrelated examples of codes. The second code has three versions. The three versions are successive changes to correct problems. With each code (code 1 and 3 versions of code 2) do the following:

1. Read the code carefully.
2. Write a short paragraph to give a high-level description of the code.
3. Look for problems in the code and describe the problems. The description should be precise and concrete. Number the problems. Label the lines of code and use those labels to refer to the code while describing the problem. Submit the labeled code along with your answers.

The homework is worth 35 points, out of which 7 points will be for the quality of writing, presentation, and following instructions. Copy the top 4 lines and fill in your name on the answer sheet. The homework must be type written.

Code 1

```
class String {
public:
    String( char *arg ); // copy constructor operator const char* ()
const;
    // ...
};

void foo( const String& aString );
void bar( const char *anArray );

// Now, we added the following class

class Word {
public:
    Word( char *arg ); // copy constructor
    // ...
};

// need another foo that works with "Words"

void foo( const Word& aWord );

int gorp()
{
    foo("hello");

    String baz = "quux";
    bar(baz);
}
```

Hints:

1. Experiment with the code by incorporating it in a complete program and running the program. Try with and without adding the class Word and the second foo().
2. Read about implicit conversion performed by compilers.
3. Read about overloading of functions.

Code 2-A: First Try

This is representative of codes written by people with at least 3 years of C++ experience. Writing the code was a part of their job interview.

```
1. #define NamedPoint1 NamedPoint
2.
3. class NamedPoint1{
4. protected:
5.     float    x;
6.     float    y;
7.     char     *name;
8. public:
9.     NamedPoint1(float x,float y,char *name){
10.         this->x=x;
11.         this->y=y;
12.         this->name=name;
13.     }
14.
15.     float getX(){return x;}
16.     float getY(){return y;}
17.     char *getName(){return name;}
18.
19.     void setX(float x){this->x=x;}
20.     void setY(float y){this->y=y;}
21.     void setName(char *name){this->name=name;}
22. };
23.
24. NamedPoint *readNamedPoint(FILE *in){
25.     char buffer[80];
26.     float x,y;
27.     NamedPoint *np;
28.     fscanf(in,"%f %f %s",&x,&y,buffer);
29.     np=new NamedPoint1(x,y,buffer);
30.     return np;
31. }
32.
33. void writeNamedPoint(NamedPoint *np,FILE *out){
34.     fprintf(out,"%f %f %s",np->getX(),np->getY(),np->getName());
35. }
36.
37. void dumpNamedPoint(FILE *in){
38.     NamedPoint *np;
39.     np=readNamedPoint(in);
40.     writeNamedPoint(np,stdout);
41.     delete np;
42. }
```

Code 2-B: Second Try

```
1. #define NamedPoint2 NamedPoint
2.
3. class NamedPoint2{
4. protected:
5.     float    x;
6.     float    y;
7.     char     *name;
8. public:
9.     NamedPoint2(float x, float y, char *name){
10.         this->x=x;
11.         this->y=y;
12.         setName(name);
13.     }
14.
15.     float getX(){return x;}
16.     float getY(){return y;}
17.     char *getName(){return name;}
18.
19.     void setX(float x){this->x=x;}
20.     void setY(float y){this->y=y;}
21.     void setName(char *name){
22.         this->name=new char[strlen(name)+1];
23.         strcpy(this->name,name);
24.     };
25. };
26.
27. void testCase1(){
28.     NamedPoint *np=new NamedPoint(1.0,1.0,"foo");
29.     np->setName("bar");
30.     delete np;
31. }
```

Code 2-C: Third Try

Many proficient engineers would tell you that they would never submit the second version, they would go right to this version. Unfortunately, this version also has some problems.

```
1. #define NamedPoint3 NamedPoint
2.
3. class NamedPoint3{
4. protected:
5.     float    x;
6.     float    y;
7.     char     *name;
8. public:
9.     NamedPoint3(float x, float y, char *name) {
10.         this->x=x;
11.         this->y=y;
12.         setName(name);
13.     }
14.
15.     ~NamedPoint3() {
16.         delete [] name;
17.     }
18.
19.     float getX() {return x;}
20.     float getY() {return y;}
21.     char *getName() {return name;}
22.
23.     void setX(float x) {this->x=x;}
24.     void setY(float y) {this->y=y;}
25.     void setName(char *name) {
26.         delete [] this->name;
27.         this->name=new char[strlen(name)+1];
28.         strcpy(this->name, name);
29.     };
30. };
31.
32. void testCase2() {
33.     NamedPoint *np1, *np2
34.     np1=new NamedPoint(1.0, 1.0, "foo");
35.     np2=new NamedPoint(1.0, 1.0, "bar");
36.
37.     *np1=*np2;
38.     delete np1;
39.     delete np2;
40. }
```