

## ECE 417/617 Assignment #2

Read the first four pages of the following paper: Michael E. Fagan, *Advances in Software Inspections*, *IEEE Transactions on Software Engineering*, 12(7):744-751, 1986. <http://www.mfagan.com/pdfs/aisi1986.pdf> .

Then answer the following questions.

1. Where and when was the software inspection process created?
2. For what two purposes was the software inspection process created?
3. Suppose that IBM's System/370 products contained 3000 defects and 1 million lines of code in 1976. By 1984, how many defects and lines of code did the products contain?
4. Does the software inspection process involve a tradeoff between cost and quality? Explain.
5. Approximately what percentage of all the defects can be expected to be found using inspections, according to the author's experience?
6. Is a trained inspection moderator necessary to achieve good results?
7. List the nine operations in the software development process.
8. How much cheaper is it to remedy defects during early operations rather than during testing or maintenance?
9. What are exit criteria? List three characteristics that exit criteria must have.
10. During a software inspection, should participants propose solutions to the defects found?
11. What is the maximum amount of time that a team should spend in a single software inspection session?
12. What is the purpose of the checklist of defect types?
13. What four roles are played by participants?
14. What is a "phantom inspector"?

15. In the following program, comment with // any lines that won't compile. Now with the offending lines removed, what is the output of this program?

```
#include <stdio>

struct Data {
    Data() : m_a(0) { printf("data con\n"); }
    ~Data() { printf("data decon\n"); }
    int m_a;
};

class FooBase {
public:
    FooBase() { printf("FooBase con\n"); }
    ~FooBase() { printf("FooBase decon\n"); }
    void Func1(const Data& d, Data* e, int f=2) {
        printf("FooBase::Func\n");
        e->m_a = d.m_a * f;
        d.m_a = 3;
    }
    virtual void Func2() {
        printf("FooBase::Func2\n");
    }
    void Func3() {
        printf("FooBase::Func3\n");
    }
private:
    int m_h;
};

class Foo : public FooBase
{
public:
    Foo() { printf("Foo con\n"); }
    ~Foo() { printf("Foo decon\n"); }
    virtual void Func2() {
        printf("Foo::Func2\n");
    }
    void Func3() {
        m_h = 2;
        printf("Foo::Func3\n");
    }
};

int main()
{
    Foo g;
    Data d1, d2;
    FooBase* h = (FooBase*) &g;
    g.Func1(d1, &d2);
    printf("m: %d %d\n", d1.m_a, d2.m_a);
    h->Func2();
    h->Func3();
    return 0;
}
```