CS 430 Scoping Lab

Consider the following program in a C-like language that allows nested procedures. For static scoping, assume that a declaration is in force from the point it appears until the end of the block in which it appears. For dynamic scoping, assume that declarations are encountered during execution in the order they appear within a procedure.

```
void main()
{

    int z = 10;

    void g()
    {
        int x = 12;
        int y = 3;                // location A

        void f()
        {
            z = x*y;
            printLine(x, y, z);   // location B
        }

        void h()
        {
            int y = 2;
            int z = 5;            // location C
            x = y+z;
            printLine(x, y, z)
            f();
        }

        h();
        printLine(x, y, z);
    }

    g();                          // location D
}
```

1. If this language has **static** scoping, what is the output of this program?

2. If this language has **dynamic** scoping, what is the output of this program?

3. If this language has **static** scoping, what is the referencing environment at each of the four locations indicated; i.e., which variables are visible and which procedures are they declared in (e.g., `f.y`, `g.y`, or `h.y`)?

Location A: _____

Location B: _____

Location C: _____

Location D: _____

4. If this language has **dynamic** scoping, what is the referencing environment at each of the four locations indicated the first time execution reaches those points?

Location A: _____

Location B: _____

Location C: _____

Location D: _____

5. Provide the following info about the variable "x" in the context of this Java code:

```
class Main {
    private int x = 100;
}
```

Name: Static Dynamic (circle one)

Address: Static Dynamic

Value: Static Dynamic

Lifetime: Static Stack-dynamic Explicit heap-dynamic Implicit heap-dynamic

Scope: _____