Name: Rylie Byers

Fill the following table with your solution:

Executable size without using –g : 472	**Why?**
Executable size when using –g : **O0 : 576** **O1: 224** **O2: 224**	-g Allows the compiler to use/generate debug information which is probably why the executable size increased. The O0, O1, and O2 represent different levels of optimization. O0 is essentially zero optimization happening. The higher the numbers the more optimization is happening. Which is why O1 is smaller. O1 and O2 are probably the same size because O2 the changes did not affect the size.

Bug #	**Details**			**Proposed solution**	
	File, line	Function name	Description	Description	Modified code
1	Function: Department.cpp,23 Bug: Department.cpp, 57	Double CalculateMaxSalary(staff types)	There was “max = SoftwareArchitects->at(i).getSalary();” Outside and under the if statement, so it would automatically override the true max value. So the last value in the software architects would probably be listed as the max salary regardless.	All you need to do to fix it is remove the extra line. And it should perform as expected.	See below*(table gets cut off)

2	Function: Department.cpp, 67 Bug: Department.cpp,77, 83 Department.cpp,67 Department.h, 34 Department.cpp, 96	Int Department CalculateAverageSalary(StaffTypes);	For lines 77 and 83 the sum was just = instead of +=. So the average salary was miscalculated. For line 67 I changed int into double because we need to return a double average. And had to update it in the header. The average wasn't as accurate as it needed to be. For line 96 for the average salary being 0 it was returning nan so I added an if statement "**if** (i == 0){ **return** 0.0; }"	Just needed to add 2 + signs. Changed int to double for the function. Added an if statement to not get "nan"	See below***(table gets cut off)
3	employeeClass.h, 30	**void** print();	The supervisor and Project ID was not printing	Just added Virtual in the header .	**virtual void** print();

* Department::SOFTWAREARCHITECTS:

```
        for (i = 0; i < SoftwareArchitects->size(); i++){
        if (SoftwareArchitects->at(i).getSalary()>max){
  max = SoftwareArchitects->at(i).getSalary();
 }
// where the extra max was
}
```

** **double** Department::CalculateAverageSalary(StaffTypes type){

```
    double sum;
        sum = 0.0;
        int i;
        switch (type){
        case Department::EMPLOYEE:
            for (i = 0; i < Employees->size(); i++){
                sum += Employees->at(i).getSalary();}
            break;
        case Department::PROGRAMMER:
            for (i = 0; i < Programmers->size(); i++){
                sum += Programmers->at(i).getSalary(); }
            break;
        case Department::SOFTWAREARCHITECTS:
            for (i = 0; i < SoftwareArchitects->size(); i++){
                sum += SoftwareArchitects->at(i).getSalary();}
            break;
        default:
            sum = -1.0;
            break;}
    if (i == 0){
       return 0.0;
    }
    return sum/i;
}
```

Add rows when necessary.