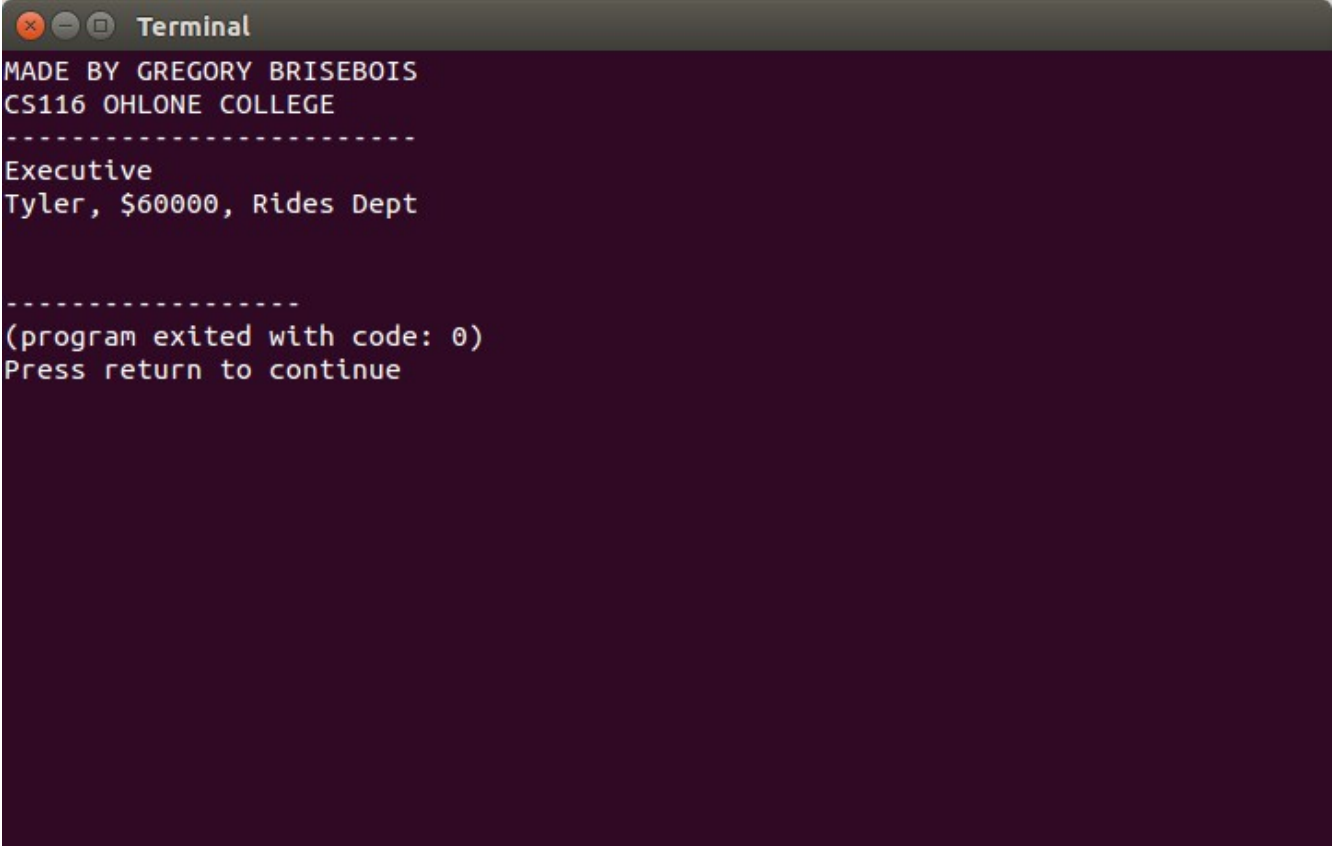


Program Output:

A screenshot of a macOS Terminal window. The title bar at the top shows the standard window controls (red, yellow, green buttons) and the word "Terminal". The terminal content is as follows:

```
MADE BY GREGORY BRISEBOIS  
CS116 OHLONE COLLEGE  
-----  
Executive  
Tyler, $60000, Rides Dept  
  
-----  
(program exited with code: 0)  
Press return to continue
```

Code:

main.cpp

```
#include <iostream>
#include <string>

using namespace std;

#include "employee.h"
#include "manager.h"
#include "executive.h"

// Course requirement - prints my own name on the
// console
void printMyName()
{
    cout << "MADE BY GREGORY BRISEBOIS" << endl;
    cout << "CS116 OHLONE COLLEGE" << endl;
    cout << "-----" << endl;
}

int main()
{
    printMyName();

    Executive myExec("Tyler", 60000, "Rides Dept");

    myExec.printInfo();
}
```

employee.h

```
class Employee
{
public:
    // Constructors
    Employee();
    Employee(string initialName, double
initialSalary);

    // Accessors
    double getSalary() const;
    string getName() const;

    // Mutators
    void setSalary(double newSalary);

private:
    // Variables
    string name;
    double salary;
};

// Default Constructor
Employee::Employee()
{
    name = "";
    salary = 0;
}

// Custom Constructor
Employee::Employee(string initialName, double
initialSalary)
{
    name = initialName;
    salary = initialSalary;
}

void Employee::setSalary(double newSalary)
{
    salary = newSalary;
}
```

```
}  
  
double Employee::getSalary() const  
{  
    return salary;  
}  
  
string Employee::getName() const  
{  
    return name;  
}
```

manager.h

```
class Manager : public Employee
{
    public:

        // Constructors
        Manager();
        Manager(string initialName, double initialSalary,
string initialDepartment);

        // Accessors
        string getDepartment() const;
        void printInfo();

        private:
        // Variables
        string department;
};

// Default Constructor
Manager::Manager() : Employee()
{
    department = "";
}

// Custom Constructor
Manager::Manager(string initialName, double
initialSalary, string initialDepartment) :
Employee(initialName, initialSalary)
{
    department = initialDepartment;
}

void Manager::printInfo()
{
    cout << getName() << ", $" << getSalary() << ", "
<< getDepartment() << endl;
}

string Manager::getDepartment() const
```

```
{  
    return department;  
}
```

executive.h

```
class Executive : public Manager
{
    public:
        // Constructors
        Executive();
        Executive(string initialName, double
initialSalary, string initialDepartment);

        void printInfo();
};

Executive::Executive() : Manager()
{
};

Executive::Executive(string initialName, double
initialSalary, string initialDepartment) :
Manager(initialName, initialSalary, initialDepartment)
{
};

void Executive::printInfo()
{
    cout << "Executive" << endl;
    Manager::printInfo();
}
```