

Forum:[Hack Forums](#)

Thread: [Anybody familiar with design patterns?](#)

Support™ - April 20, 2016, 1:16 am

Hey guys! There is a part of my assignment that I need to implement a design pattern in the project I am working on. It is a simple data management software which has only one class(Employee class) and a form for performing Create, Read, Update and Delete(CRUD) operations.

Here is my Employee class:

Code using namespace System;
using namespace System::IO;

```
ref class Employee
{
private:
    String^ employeeID;
    String^ employeeFirstName;
    String^ employeeLastName;
    String^ employeeEmailAddress;

public:
    Employee(String^ employeeID){
        this->employeeID = employeeID;
    }
public:
    Employee();

    String^ get_employeeFirstName(){
        return employeeFirstName;
    }
    String^ get_employeeLastName(){
        return employeeLastName;
    }
    String^ get_employeeEmailAddress(){
        return employeeEmailAddress;
    }
    void set_employeeFirstName(String^ employeeFirstName){
        this->employeeFirstName = employeeFirstName;
    }
    void set_employeeLastName(String^ employeeLastName){
        this->employeeLastName = employeeLastName;
    }
    void set_employeeEmailAddress(String^ employeeEmailAddress){
        this->employeeEmailAddress = employeeEmailAddress;
    }

public:
```

```

void createEmployeeRecord(String^ employeePath){
    //Writes to a file
    //IO::File::WriteAllText(employeePath, employeeData);
}
public:
void getEmployeeData(String^ employeeDataPath){
    //String^ employeeData = IO::File::ReadAllText(employeeDataPath);
    //Reads from a file
}
};

```

Then in the form operations, I use this method to create employee records:

```

Code public: System::Void createEmployee( String^ employeeID, String^ firstName, String^
lastName, String^ emailAddress){

```

```

    Employee^ employee = gcnew Employee(employeeID);
    employee->set_employeeFirstName(firstName);
    employee->set_employeeLastName(lastName);
    employee->set_employeeEmailAddress(emailAddress);

```

```

employee->createEmployeeRecord(employeePath);
}

```

What design pattern does this fall under?

I'm suspecting that it is either Builder pattern or Singleton pattern.

If not, can you enlighten me about what has to be changed in the above class and method to make it one of these two patterns (or) which is the easiest design pattern to follow if I have rebuild this class from ground up? It is not a complex assignment, it ends right here.

0xDEAD10CC - April 20, 2016, 2:25 am

Why does this class have multiple public labels? It makes zero sense, and why would it be a singleton if this is the case:

Code public:

```
Employee();
```

Public default constructor, accessible copy constructors... None of which a singleton by definition would allow.

Support™ - April 20, 2016, 2:34 am

(Apr 20, 2016 02:25 AM)0xDEAD10CC Wrote: [\[CITE\]](#) Why does this class have multiple public labels? It makes zero sense, and why would it be a singleton if this is the case:

Code public:

```
Employee();
```

Public default constructor, accessible copy constructors... None of which a singleton by definition would allow.

Found that out after reading more about design patterns. Currently, it is safe to say that I don't know much about these. Closing the thread to start reading :D

I have found a nice place which explains about design patterns and hope it helps someone else too. http://www.tutorialspoint.com/design_pattern/index.htm

(Although it's in Java, I can understand the concepts)