The Façade and Factory Method Pattern

Design Patterns

Mark Swarner

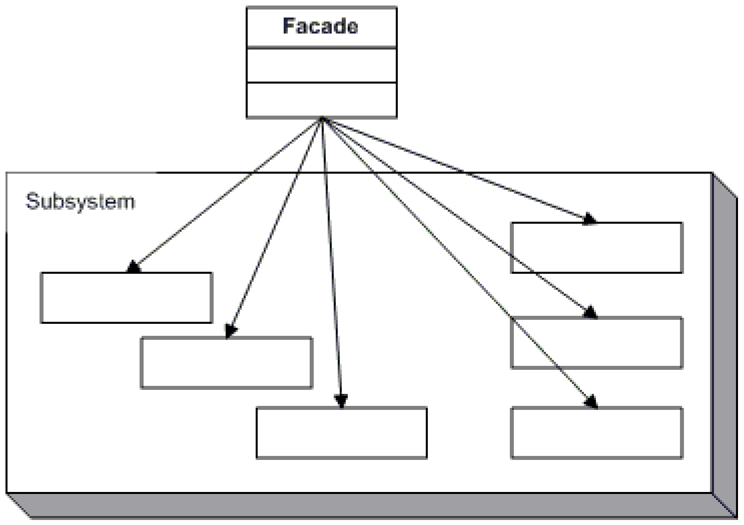
Fall 2016-17

Introduction:

For this assignment, we were required to write a program in C# that correctly implements the Façade and/or the Factory Method patterns. My submission allows for a programmer to manage multiple open projects from a single program with minimal effort on the programmer's side.

The UML Diagram for the Façade Pattern:

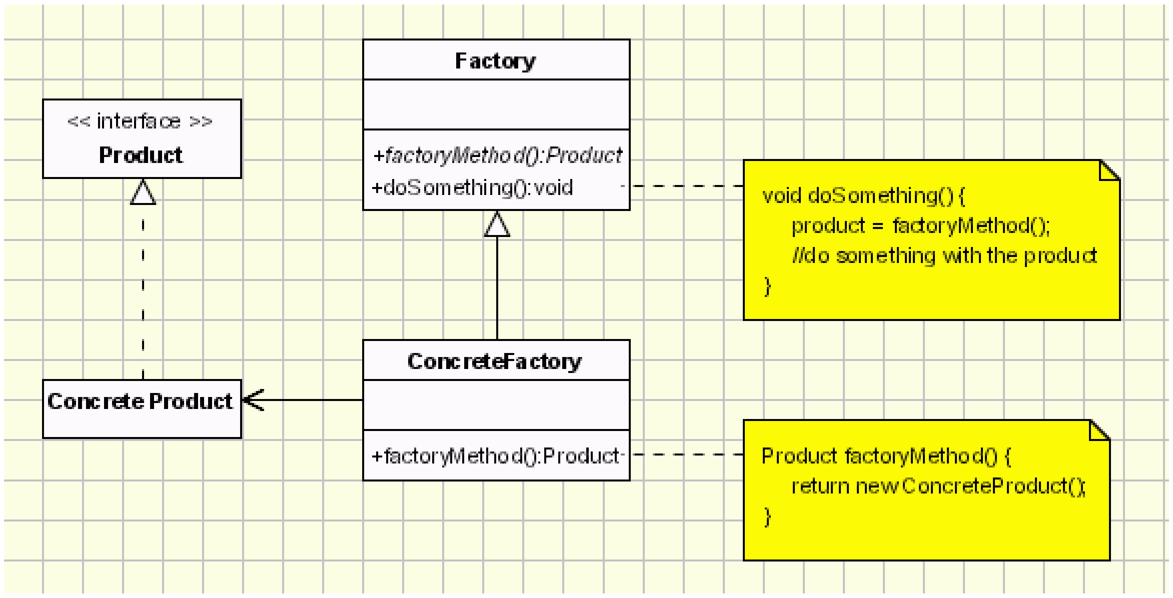
The UML Diagram shown to the right illustrates the basic structure of a program utilizing the Façade pattern. It also describes the individual classes and functions. The table below explains how I utilized the structure:



Facade	The Façade Class controls multiple subsystems that exist outside of the program.
PcmForm	PcmForm serves as a client that utilizes the Façade to manage the subsystems.

The UML Diagram for the Factory Method Pattern:

The UML Diagram shown to the right illustrates the basic structure of a program utilizing the Factory Method. It also describes the individual classes and functions. The table below explains how I utilized the structure:



```
void doSomething() {
    product = factoryMethod();
    //do something with the product
}
```

```
Product factoryMethod() {
    return new ConcreteProduct();
}
```

PcmForm	Instantiates the Factory Method Pattern
Factory	Contains the instructions for the “doSomething()” function and acts as an abstract parent for “ConcreteFactory”
ConcreteFactory	Concrete Factory contains instructions for the “factoryMethod()” function, whose signature was defined within “Factory”
Product	“Product” is the object that is created by “Factory”. It is defined as a form with the necessary objects for it to function
ConcreteProduct	Is a child of “Product” that directly creates each and every Instance of “Product”

Narrative:

```
public class Facade
    {
        String visualStudioPath = "C:/Program Files (x86)/Microsoft Visual Studio 
12.0/Common7/IDE/devenv.exe";
        String eclipsePath = "C:/Users/Mark/eclipse/java-neon/eclipse/eclipse.exe";
        String sublimeTextPath = "C:/Program Files/Sublime Text 3/sublime_text.exe";
        String bracketsPath = "C:/Program Files (x86)/Brackets/Brackets.exe";
        String netbeansPath = "C:/Program Files/NetBeans 8.1/bin/netbeans64.exe";
        String phpStormPath = "C:/Program Files (x86)/JetBrains/PhpStorm 
10.0.3/bin/PhpStorm.exe";
        String chromePath = "C:/Program Files 
(x86)/Google/Chrome/Application/chrome.exe";
        String edgePath = 
"C:/Windows/SystemApps/Microsoft.MicrosoftEdge_8wekyb3d8bbwe/MicrosoftEdge.exe";
        String firefoxPath = "C:/Program Files (x86)/Mozilla Firefox/firefox.exe";
        String operaPath = "C:/Program Files (x86)/Opera/launcher.exe";
        String waterfoxPath = "C:/Program Files/Waterfox/waterfox.exe";
        String cyberduckPath = "C:/Program Files (x86)/Cyberduck/Cyberduck.exe";
        String xftpPath = "C:/Program Files (x86)/NetSarang/Xftp 5/Xftp.exe";
        String cmdPath = "C:/Windows/system32/cmd.exe";
        String powerShellPath = 
"C:/Windows/System32/WindowsPowerShell/v1.0/powershell.exe";
        String xshellPath = "C:/Program Files (x86)/NetSarang/Xshell 5/Xshell.exe";
        String githubDesktopPath = 
"C:/Users/Mark/AppData/Local/Apps/2.0/VYWTXCG6.7QZ/A3DMKXJL.EC1/gith..tion_317444273a93
ac29_0003.0003_92b520eb113e6614/GitHub.exe";
        String mySQLServerPath = "C:/Program Files/MySQL/MySQL Workbench 6.3 
CE/MySQLWorkbench.exe";
        String xamppPath = "C:/xampp/xampp-control.exe";
        Process ideProcess;
        int ideKillInt;
        Process browserProcess;
        int browserKillInt;
        Process ftpClientProcess;
        int ftpClientKillInt;
        Process shellInterfaceProcess;
        int shellInterfaceKillInt;
        Process githubDesktopProcess;
        int githubDesktopKillInt;
        Process mySQLServerProcess;
        int mySQLServerKillInt;
        Process xamppProcess;
        int xamppKillInt;
        public Facade()
        {

        }

        public Facade(String projectType, String ide, String browser, String ftpClient, 
String shellInterface, Boolean githubDesktop, Boolean mySQLServer, Boolean xampp)
        {
            if (ide == "Brackets")
            {
                ideProcess = Process.Start(bracketsPath);
                ideKillInt = Process.GetProcessesByName(ideProcess.ProcessName).Length;
            }
            else if (ide == "Eclipse")
            {
                ideProcess = Process.Start(eclipsePath);
                ideKillInt = Process.GetProcessesByName(ideProcess.ProcessName).Length;
            }
            else if (ide == "Netbeans")
            {
                ideProcess = Process.Start(netbeansPath);
                ideKillInt = Process.GetProcessesByName(ideProcess.ProcessName).Length;
            }
            else if (ide == "PHPStorm")
            {
                ideProcess = Process.Start(phpStormPath);
                ideKillInt = Process.GetProcessesByName(ideProcess.ProcessName).Length;
            }
            else if (ide == "Sublime Text")
            {
                ideProcess = Process.Start(sublimeTextPath);
                ideKillInt = Process.GetProcessesByName(ideProcess.ProcessName).Length;
            }
            else if (ide == "Visual Studio")
            {
                ideProcess = Process.Start(visualStudioPath);
                ideKillInt = Process.GetProcessesByName(ideProcess.ProcessName).Length;
            }
            if (browser == "Chrome")
            {
                browserProcess = Process.Start(chromePath);
                browserKillInt = 
Process.GetProcessesByName(browserProcess.ProcessName).Length;
            }
            else if (browser == "Edge")
            {
                browserProcess = Process.Start(edgePath);
                browserKillInt = 
Process.GetProcessesByName(browserProcess.ProcessName).Length;
            }
            else if (browser == "Firefox")
            {
                browserProcess = Process.Start(firefoxPath);
                browserKillInt =
Process.GetProcessesByName(browserProcess.ProcessName).Length;
            }
            else if (browser == "Opera")
            {
                browserProcess = Process.Start(operaPath);
                browserKillInt =
Process.GetProcessesByName(browserProcess.ProcessName).Length;
            }
            else if (browser == "Waterfox")
            {
                browserProcess = Process.Start(waterfoxPath);
                browserKillInt =
Process.GetProcessesByName(browserProcess.ProcessName).Length;
            }
            if (ftpClient == "Cyberduck")
            {
                ftpClientProcess = Process.Start(cyberduckPath);
                ftpClientKillInt =
Process.GetProcessesByName(ftpClientProcess.ProcessName).Length;
            }
            else if (ftpClient == "Xftp")
            {
                ftpClientProcess = Process.Start(xftpPath);
                ftpClientKillInt =
Process.GetProcessesByName(ftpClientProcess.ProcessName).Length;
            }
            if (shellInterface == "Command Prompt")
            {
                shellInterfaceProcess = Process.Start(cmdPath);
                shellInterfaceKillInt =
Process.GetProcessesByName(shellInterfaceProcess.ProcessName).Length;
            }
            else if (shellInterface == "Powershell")
            {
                shellInterfaceProcess = Process.Start(powerShellPath);
                shellInterfaceKillInt =
Process.GetProcessesByName(shellInterfaceProcess.ProcessName).Length;
            }
            else if (shellInterface == "Xshell")
            {
                shellInterfaceProcess = Process.Start(xshellPath);
                shellInterfaceKillInt =
Process.GetProcessesByName(shellInterfaceProcess.ProcessName).Length;
            }
            if (githubDesktop)
            {
                githubDesktopProcess = Process.Start(githubDesktopPath);
                githubDesktopKillInt =
Process.GetProcessesByName(githubDesktopProcess.ProcessName).Length;
            }
            if (mySQLServer)
            {
                mySQLServerProcess = Process.Start(mySQLServerPath);
                mySQLServerKillInt =
Process.GetProcessesByName(mySQLServerProcess.ProcessName).Length;
            }
            if (xampp)
            {
                xamppProcess = Process.Start(xamppPath);
                xamppKillInt =
Process.GetProcessesByName(xamppProcess.ProcessName).Length;
            }
        }
        public void murder(){
            if (ideKillInt != 0)
            {
                foreach (Process proc in
Process.GetProcessesByName(ideProcess.ProcessName))
                {
                    proc.Kill();
                }
            }
            if (ftpClientKillInt != 0)
            {
                foreach (Process proc in
Process.GetProcessesByName(ftpClientProcess.ProcessName))
                {
                    proc.Kill();
                }
            }
            if (shellInterfaceKillInt != 0)
            {
                foreach (Process proc in
Process.GetProcessesByName(shellInterfaceProcess.ProcessName))
                {
                    proc.Kill();
                }
            }
            if (githubDesktopKillInt != 0)
            {
                foreach (Process proc in
Process.GetProcessesByName(githubDesktopProcess.ProcessName))
                {
                    proc.Kill();
                }
            }
            if (mySQLServerKillInt != 0)
            {
                foreach (Process proc in
Process.GetProcessesByName(mySQLServerProcess.ProcessName))
                {
                    proc.Kill();
                }
            }
            if (xamppKillInt != 0)
            {
                foreach (Process proc in
Process.GetProcessesByName(xamppProcess.ProcessName))
                {
                    proc.Kill();
                }
            }
        }
    }
```

The “Façade” class has a few pieces to it. The first piece of code defines a multitude of strings, all of which are placeholders for the paths of each necessary file. The next chunk creates a multitude of processes and integers. These are used for starting and killing processes later. The constructor for “Façade” is created next. The constructor is overloaded, containing the necessary function, which takes zero arguments and has no code within its body. There is also a version that takes multiple arguments that are used in the if else if statements to determine which processes to create and then defines the integers used to determine if a process needs to be killed. This is the version the program uses to implement the Façade pattern. Following the constructor is the “murder()” function, which uses the previously defined integers to determine if a process is running and if so, ends that process.

```
abstract class Factory
    {
        public abstract Product factoryMethod(String projectName, String projectType,
String ide, String browser, String ftpClient, String shellInterface, Boolean
githubDesktop, Boolean mySQLServer, Boolean xampp);

        public void doSomething(String projectName, String projectType, String ide,
String browser, String ftpClient, String shellInterface, Boolean githubDesktop, Boolean
mySQLServer, Boolean xampp)
        {
            Product product = factoryMethod(projectName,  projectType,  ide,  browser,
ftpClient,  shellInterface,  githubDesktop,  mySQLServer,  xampp);
        }
    }
```

The “Factory” class contains two methods, but only has instructions for one. The “doSomething()” function has arguments, but are only used to send variables to satisfy the arguments of the “factoryMethod()” function. The body of the function creates a Product called product ad defines it with the “factoryMethod()” function.

```
class ConcreteFactory:Factory
    {
        public ConcreteProduct cp;
        public ConcreteFactory()
        {

        }
        public override Product factoryMethod(String projectName, String projectType,
String ide, String browser, String ftpClient, String shellInterface, Boolean
githubDesktop, Boolean mySQLServer, Boolean xampp)
        {
            return cp = new ConcreteProduct(projectName, projectType, ide, browser,
ftpClient, shellInterface, githubDesktop, mySQLServer, xampp);
        }
    }
```

The “ConcreteFactory” class contains an empty constructor to allow for instantiation. Following the constructor is the “factoryMethod()” function which takes multiple arguments that are used to call the “ConcreteProduct()” method in the return statement.

```
public partial class Product : Form
    {
        public Product()
        {

        }
        public Product(String projectName, String projectType, String ide, String
browser, String ftpClient, String shellInterface, Boolean githubDesktop, Boolean
mySQLServer, Boolean xampp)
        {
            InitializeComponent();
            Text = projectName;
            infoBox.Items.Add(projectName);
            infoBox.Items.Add(projectType);
            infoBox.Items.Add(ide);
            infoBox.Items.Add(browser);
            infoBox.Items.Add(ftpClient);
            infoBox.Items.Add(shellInterface);
            infoBox.Items.Add(Convert.ToString(githubDesktop));
            infoBox.Items.Add(Convert.ToString(mySQLServer));
            infoBox.Items.Add(Convert.ToString(xampp));
        }
    }
```

The "Product" class is a Form that contains an empty Constructor and an overloaded constructor with multiple arguments. The body of the constructor calls the "InitializeComponents()" function, which is created and defined by Visual Studio. It then uses the "projectName" string to title the form. The rest of the body just takes the variables instantiated by the arguments and populates the "infoBox" listbox with the string values of those variables.

```
class ConcreteProduct:Product
    {
        public Product project;
        public ConcreteProduct(String projectName, String projectType, String ide,
String browser, String ftpClient, String shellInterface, Boolean githubDesktop, Boolean
mySQLServer, Boolean xampp)
        {
            project = new Product(projectName, projectType, ide, browser, ftpClient,
shellInterface, githubDesktop, mySQLServer, xampp);
            project.Visible = true;
        }
    }
```

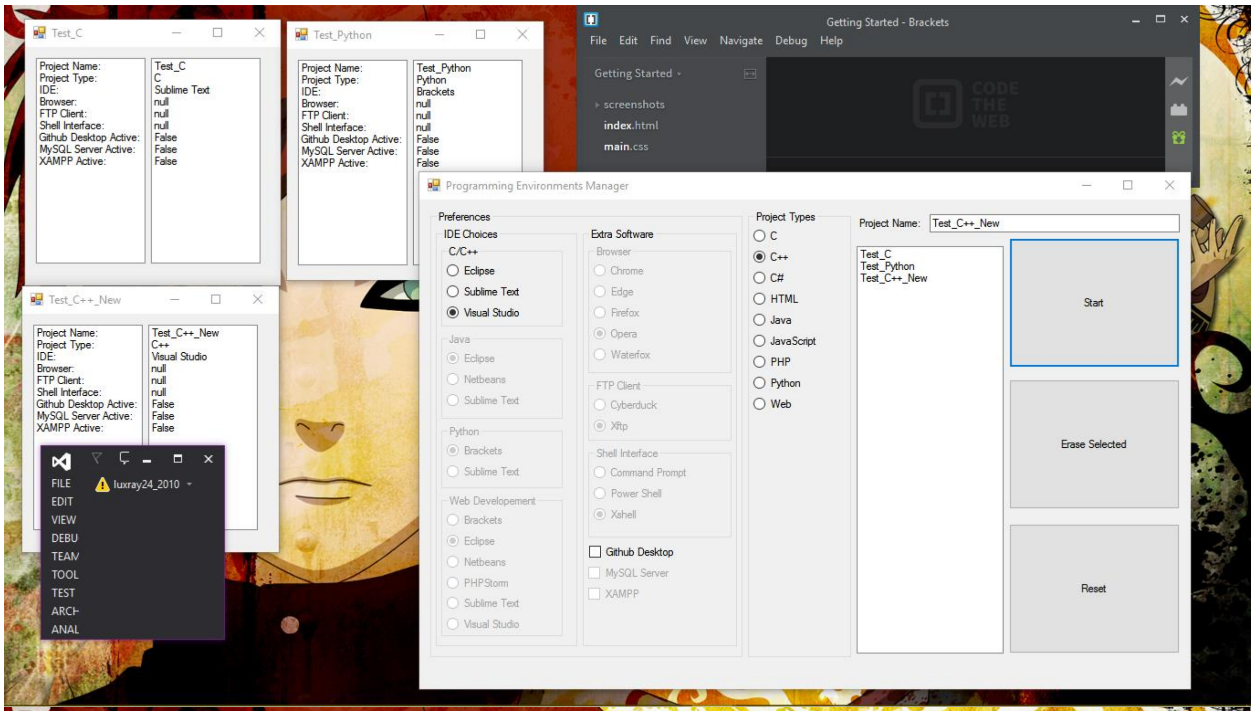
The "ConcreteProduct" class contains only a constructor, which takes arguments, and creates an instance of the "project" Product, passing arguments to Product.

```
public partial class pcmForm : Form
    {
        List<Facade> facade = new List<Facade>();
        List<Form> openForms = new List<Form>();
        ConcreteFactory fmp = new ConcreteFactory();
        String projectName = "";
        String projectType = "C#";
        String ide = "Visual Studio";
        String browser = "null";
        String ftpClient = "null";
        String shellInterface = "null";
        Boolean githubDesktop = true;
        Boolean mySQLServer = false;
        Boolean xampp = false;
        public pcmForm()
        {
            InitializeComponent();
            Name = "pcmForm";
        }

        private void pcButton_Click(object sender, EventArgs e)
        {
            projectName = projectNameTextBox.Text;
            projectListBox.Items.Add(projectName);
            if (projectListBox.Items.Count == 1)
            {
                projectListBox.SelectedIndex = projectListBox.FindString(projectName);
            }
            facade.Add(new Facade(projectType, ide, browser, ftpClient, shellInterface,
githubDesktop, mySQLServer, xampp));
            fmp.doSomething(projectName, projectType, ide, browser, ftpClient,
shellInterface, githubDesktop, mySQLServer, xampp);
            openForms.Add(fmp.cp.project);
        }

        private void cRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (cRadio.Checked == true)
            {
                cBox.Enabled = true;
                projectType = "C";
                if (eclipseCRadio.Checked == true)
                {
                    ide = "Eclipse";
                }
                else if (sublimeTextCRadio.Checked == true)
                {
                    ide = "Sublime Text";
                }
                else if (visualStudioCRadio.Checked == true)
                {
                    ide = "Visual Studio";
                }
                browser = "null";
                ftpClient = "null";
                shellInterface = "null";
                if (githubCheck.Checked == true)
                {
                    githubDesktop = true;
                }
                else
                {
                    githubDesktop = false;
                }
                mySQLServer = false;
                xampp = false;
            }
            else if (cRadio.Checked == false)
            {
                cBox.Enabled = false;
            }
        }

        private void cppRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (cppRadio.Checked == true)
            {
                cBox.Enabled = true;
                projectType = "C++";
                if (eclipseCRadio.Checked == true)
                {
                    ide = "Eclipse";
                }
                else if (sublimeTextCRadio.Checked == true)
                {
                    ide = "Sublime Text";
                }
                else if (visualStudioCRadio.Checked == true)
                {
                    ide = "Visual Studio";
                }
                browser = "null";
                ftpClient = "null";
                shellInterface = "null";
                if (githubCheck.Checked == true)
                {
                    githubDesktop = true;
                }
                else
                {
                    githubDesktop = false;
                }
                mySQLServer = false;
                xampp = false;
            }
            else if (cppRadio.Checked == false)
            {
                cBox.Enabled = false;
            }
        }

        private void csRadio_CheckedChanged(object sender, EventArgs e)
        {
            projectType = "C#";
            ide = "Visual Studio";
            browser = "null";
            ftpClient = "null";
            shellInterface = "null";
            if (githubCheck.Checked == true)
            {
                githubDesktop = true;
            }
            else
            {
                githubDesktop = false;
            }
            mySQLServer = false;
            xampp = false;
        }

        private void htmlRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (htmlRadio.Checked == true)
            {
                webIDEBox.Enabled = true;
                browserBox.Enabled = true;
                ftpBox.Enabled = true;
                shellInterfaceBox.Enabled = true;
                mySQLCheck.Enabled = true;
                xamppCheck.Enabled = true;
                projectType = "HTML";
                if (bracketsRadio.Checked == true)
                {
                    ide = "Brackets";
                }
                else if (eclipseWebRadio.Checked == true)
                {
                    ide = "Eclipse";
                }
                else if (netbeansWebRadio.Checked == true)
                {
                    ide = "Netbeans";
                }
                else if (phpStormRadio.Checked == true)
                {
                    ide = "PHPStorm";
                }
                else if (sublimeTextWebRadio.Checked == true)
                {
                    ide = "Sublime Text";
                }
                else if (visualStudioWebRadio.Checked == true)
                {
                    ide = "Visual Studio";
                }
                if (chromeRadio.Checked == true)
                {
                    browser = "Chrome";
                }
                else if (edgeRadio.Checked == true)
                {
                    browser = "Edge";
                }
                else if (firefoxRadio.Checked == true)
                {
                    browser = "Firefox";
                }
                else if (operaRadio.Checked == true)
                {
                    browser = "Opera";
                }
                else if (waterfoxRadio.Checked == true)
                {
                    browser = "Waterfox";
                }
                if (cyberduckRadio.Checked == true)
                {
                    ftpClient = "Cyberduck";
                }
                else if (xFTPRadio.Checked == true)
                {
                    ftpClient = "Xftp";
                }
                if (cmdRadio.Checked == true)
                {
                    shellInterface = "Command Prompt";
                }
                else if (powerShellRadio.Checked == true)
                {
                    shellInterface = "Powershell";
                }
                else if (xShellRadio.Checked == true)
                {
                    shellInterface = "Xshell";
                }
                if (githubCheck.Checked == true)
                {
                    githubDesktop = true;
                }
                else
                {
                    githubDesktop = false;
                }
                if (mySQLCheck.Checked == true)
                {
                    mySQLServer = true;
                }
                else
                {
                    mySQLServer = false;
                }
                if (xamppCheck.Checked == true)
                {
                    xampp = true;
                }
                else
                {
                    xampp = false;
                }
            }
            else if (htmlRadio.Checked == false)
            {
                webIDEBox.Enabled = false;
                browserBox.Enabled = false;
                ftpBox.Enabled = false;
                shellInterfaceBox.Enabled = false;
                mySQLCheck.Enabled = false;
                xamppCheck.Enabled = false;
            }
        }

        private void javaRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (javaRadio.Checked == true)
            {
                javaIDEBox.Enabled = true;
                projectType = "C++";
                if (eclipseJavaRadio.Checked == true)
                {
                    ide = "Eclipse";
                }
                else if (netbeansJavaRadio.Checked == true)
                {
                    ide = "Netbeans";
                }
                else if (sublimeTextJavaRadio.Checked == true)
                {
                    ide = "Sublime Text";
                }
                browser = "null";
                ftpClient = "null";
                shellInterface = "null";
                if (githubCheck.Checked == true)
                {
                    githubDesktop = true;
                }
                else
                {
                    githubDesktop = false;
                }
                mySQLServer = false;
                xampp = false;
            }
            else if (javaRadio.Checked == false)
            {
                javaIDEBox.Enabled = false;
            }
        }

        private void jsRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (jsRadio.Checked == true)
            {
                webIDEBox.Enabled = true;
                browserBox.Enabled = true;
                ftpBox.Enabled = true;
                shellInterfaceBox.Enabled = true;
                mySQLCheck.Enabled = true;
                xamppCheck.Enabled = true;
                projectType = "JavaScript";
                if (bracketsRadio.Checked == true)
                {
                    ide = "Brackets";
                }
                else if (eclipseWebRadio.Checked == true)
                {
                    ide = "Eclipse";
                }
                else if (netbeansWebRadio.Checked == true)
                {
                    ide = "Netbeans";
                }
```

```
else if (phpStormRadio.Checked == true)
{
    ide = "PHPStorm";
}
else if (sublimeTextWebRadio.Checked == true)
{
    ide = "Sublime Text";
}
else if (visualStudioWebRadio.Checked == true)
{
    ide = "Visual Studio";
}
if (chromeRadio.Checked == true)
{
    browser = "Chrome";
}
else if (edgeRadio.Checked == true)
{
    browser = "Edge";
}
else if (firefoxRadio.Checked == true)
{
    browser = "Firefox";
}
else if (operaRadio.Checked == true)
{
    browser = "Opera";
}
else if (waterfoxRadio.Checked == true)
{
    browser = "Waterfox";
}
if (cyberduckRadio.Checked == true)
{
    ftpClient = "Cyberduck";
}
else if (xFTPRadio.Checked == true)
{
    ftpClient = "Xftp";
}
if (cmdRadio.Checked == true)
{
    shellInterface = "Command Prompt";
}
else if (powerShellRadio.Checked == true)
{
    shellInterface = "Powershell";
}
else if (xShellRadio.Checked == true)
{
    shellInterface = "Xshell";
}
if (githubCheck.Checked == true)
{
    githubDesktop = true;
}
else
{
```

```
                    githubDesktop = false;
                }
                if (mySQLCheck.Checked == true)
                {
                    mySQLServer = true;
                }
                else
                {
                    mySQLServer = false;
                }
                if (xamppCheck.Checked == true)
                {
                    xampp = true;
                }
                else
                {
                    xampp = false;
                }
            }
            else if (jsRadio.Checked == false)
            {
                webIDEBox.Enabled = false;
                browserBox.Enabled = false;
                ftpBox.Enabled = false;
                shellInterfaceBox.Enabled = false;
                mySQLCheck.Enabled = false;
                xamppCheck.Enabled = false;
            }
        }

        private void phpRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (phpRadio.Checked == true)
            {
                webIDEBox.Enabled = true;
                browserBox.Enabled = true;
                ftpBox.Enabled = true;
                shellInterfaceBox.Enabled = true;
                mySQLCheck.Enabled = true;
                xamppCheck.Enabled = true;
                projectType = "PHP";
                if (bracketsRadio.Checked == true)
                {
                    ide = "Brackets";
                }
                else if (eclipseWebRadio.Checked == true)
                {
                    ide = "Eclipse";
                }
                else if (netbeansWebRadio.Checked == true)
                {
                    ide = "Netbeans";
                }
                else if (phpStormRadio.Checked == true)
                {
                    ide = "PHPStorm";
                }
                else if (sublimeTextWebRadio.Checked == true)
                {
                    ide = "Sublime Text";
                }
                else if (visualStudioWebRadio.Checked == true)
                {
                    ide = "Visual Studio";
                }
                if (chromeRadio.Checked == true)
                {
                    browser = "Chrome";
                }
                else if (edgeRadio.Checked == true)
                {
                    browser = "Edge";
                }
                else if (firefoxRadio.Checked == true)
                {
                    browser = "Firefox";
                }
                else if (operaRadio.Checked == true)
                {
                    browser = "Opera";
                }
                else if (waterfoxRadio.Checked == true)
                {
                    browser = "Waterfox";
                }
                if (cyberduckRadio.Checked == true)
                {
                    ftpClient = "Cyberduck";
                }
                else if (xFTPRadio.Checked == true)
                {
                    ftpClient = "Xftp";
                }
                if (cmdRadio.Checked == true)
                {
                    shellInterface = "Command Prompt";
                }
                else if (powerShellRadio.Checked == true)
                {
                    shellInterface = "Powershell";
                }
                else if (xShellRadio.Checked == true)
                {
                    shellInterface = "Xshell";
                }
                if (githubCheck.Checked == true)
                {
                    githubDesktop = true;
                }
                else
                {
                    githubDesktop = false;
                }
                if (mySQLCheck.Checked == true)
                {
                    mySQLServer = true;
                }
                else
                {
                    mySQLServer = false;
                }
                if (xamppCheck.Checked == true)
                {
                    xampp = true;
                }
                else
                {
                    xampp = false;
                }
            }
            else if (phpRadio.Checked == false)
            {
                webIDEBox.Enabled = false;
                browserBox.Enabled = false;
                ftpBox.Enabled = false;
                shellInterfaceBox.Enabled = false;
                mySQLCheck.Enabled = false;
                xamppCheck.Enabled = false;
            }
        }

        private void pythonRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (pythonRadio.Checked == true)
            {
                pythonBox.Enabled = true;
                projectType = "Python";
                if (bracketsPythonRadio.Checked == true)
                {
                    ide = "Brackets";
                }
                else if (sublimeTextPythonRadio.Checked == true)
                {
                    ide = "Sublime Text";
                }
                browser = "null";
                ftpClient = "null";
                shellInterface = "null";
                if (githubCheck.Checked == true)
                {
                    githubDesktop = true;
                }
                else
                {
                    githubDesktop = false;
                }
                mySQLServer = false;
                xampp = false;
            }
            else if (pythonRadio.Checked == false)
            {
                pythonBox.Enabled = false;
            }
        }
        private void webRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (webRadio.Checked == true)
            {
                webIDEBox.Enabled = true;
                browserBox.Enabled = true;
                ftpBox.Enabled = true;
                shellInterfaceBox.Enabled = true;
                mySQLCheck.Enabled = true;
                xamppCheck.Enabled = true;
                projectType = "Web";
                if (bracketsRadio.Checked == true)
                {
                    ide = "Brackets";
                }
                else if (eclipseWebRadio.Checked == true)
                {
                    ide = "Eclipse";
                }
                else if (netbeansWebRadio.Checked == true)
                {
                    ide = "Netbeans";
                }
                else if (phpStormRadio.Checked == true)
                {
                    ide = "PHPStorm";
                }
                else if (sublimeTextWebRadio.Checked == true)
                {
                    ide = "Sublime Text";
                }
                else if (visualStudioWebRadio.Checked == true)
                {
                    ide = "Visual Studio";
                }
                if (chromeRadio.Checked == true)
                {
                    browser = "Chrome";
                }
                else if (edgeRadio.Checked == true)
                {
                    browser = "Edge";
                }
                else if (firefoxRadio.Checked == true)
                {
                    browser = "Firefox";
                }
                else if (operaRadio.Checked == true)
                {
                    browser = "Opera";
                }
                else if (waterfoxRadio.Checked == true)
                {
                    browser = "Waterfox";
                }
                if (cyberduckRadio.Checked == true)
                {
                    ftpClient = "Cyberduck";
                }
                else if (xFTPRadio.Checked == true)
                {
                    ftpClient = "Xftp";
                }
                if (cmdRadio.Checked == true)
                {
                    shellInterface = "Command Prompt";
                }
                else if (powerShellRadio.Checked == true)
                {
                    shellInterface = "Powershell";
                }
                else if (xShellRadio.Checked == true)
                {
                    shellInterface = "Xshell";
                }
                if (githubCheck.Checked == true)
                {
                    githubDesktop = true;
                }
                else
                {
                    githubDesktop = false;
                }
                if (mySQLCheck.Checked == true)
                {
                    mySQLServer = true;
                }
                else
                {
                    mySQLServer = false;
                }
                if (xamppCheck.Checked == true)
                {
                    xampp = true;
                }
                else
                {
                    xampp = false;
                }
            }
            else if (webRadio.Checked == false)
            {
                webIDEBox.Enabled = false;
                browserBox.Enabled = false;
                ftpBox.Enabled = false;
                shellInterfaceBox.Enabled = false;
                mySQLCheck.Enabled = false;
                xamppCheck.Enabled = false;
            }
        }

        private void githubCheck_CheckedChanged(object sender, EventArgs e)
        {
            if (githubCheck.Checked == true)
            {
                githubDesktop = true;
            }
            else
            {
                githubDesktop = false;
            }
        }

        private void mySQLCheck_CheckedChanged(object sender, EventArgs e)
        {
            if (mySQLCheck.Checked == true)
            {
                mySQLServer = true;
            }
            else
            {
                mySQLServer = false;
            }
        }

        private void xamppCheck_CheckedChanged(object sender, EventArgs e)
        {
            if (xamppCheck.Checked == true)
            {
                xampp = true;
            }
            else
            {
                xampp = false;
            }
        }

        private void eclipseCRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (eclipseCRadio.Checked == true)
            {
                ide = "Eclipse";
            }
        }

        private void sublimeTextCRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (sublimeTextCRadio.Checked == true)
            {
                ide = "Sublime Text";
            }
        }

        private void visualStudioCRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (visualStudioCRadio.Checked == true)
            {
                ide = "Visual Studio";
            }
        }

        private void bracketsRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (bracketsRadio.Checked == true)
            {
                ide = "Brackets";
            }
        }

        private void eclipseWebRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (eclipseWebRadio.Checked == true)
            {
                ide = "Eclipse";
            }
        }

        private void netbeansWebRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (netbeansWebRadio.Checked == true)
            {
                ide = "Netbeans";
            }
        }

        private void phpStormRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (phpStormRadio.Checked == true)
            {
                ide = "PHPStorm";
            }
        }

        private void sublimeTextWebRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (sublimeTextWebRadio.Checked == true)
            {
                ide = "Sublime Text";
            }
        }

        private void visualStudioWebRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (visualStudioWebRadio.Checked == true)
            {
                ide = "Visual Studio";
            }
        }

        private void chromeRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (chromeRadio.Checked == true)
            {
                browser = "Chrome";
            }
        }

        private void edgeRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (edgeRadio.Checked == true)
            {
                browser = "Edge";
            }
        }

        private void firefoxRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (firefoxRadio.Checked == true)
            {
                browser = "Firefox";
            }
        }

        private void operaRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (operaRadio.Checked == true)
            {
                browser = "Opera";
            }
        }

        private void waterfoxRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (waterfoxRadio.Checked == true)
            {
                browser = "Waterfox";
            }
        }

        private void cyberduckRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (cyberduckRadio.Checked == true)
            {
                ftpClient = "Cyberduck";
            }
        }

        private void xFTPRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (xFTPRadio.Checked == true)
            {
                ftpClient = "Xftp";
            }
        }

        private void cmdRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (cmdRadio.Checked == true)
            {
                shellInterface = "Command Prompt";
            }
        }

        private void powerShellRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (powerShellRadio.Checked == true)
            {
                shellInterface = "Powershell";
            }
        }

        private void xShellRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (xShellRadio.Checked == true)
            {
                shellInterface = "Xshell";
            }
        }

        private void eclipseJavaRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (eclipseJavaRadio.Checked == true)
            {
                ide = "Eclipse";
            }
        }

        private void netbeansJavaRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (netbeansJavaRadio.Checked == true)
            {
                ide = "Netbeans";
            }
        }

        private void sublimeTextJavaRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (sublimeTextJavaRadio.Checked == true)
            {
                ide = "Sublime Text";
            }
        }

        private void bracketsPythonRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (bracketsPythonRadio.Checked == true)
            {
                ide = "Brackets";
            }
        }

        private void sublimeTextPythonRadio_CheckedChanged(object sender, EventArgs e)
        {
            if (sublimeTextPythonRadio.Checked == true)
            {
                ide = "Sublime Text";
            }
        }

        private void resetButton_Click(object sender, EventArgs e)
        {
            projectListBox.Items.Clear();

            foreach (Form f in openForms)
            {
                f.Close();
            }
            openForms.Clear();

            for (int i = 0; i < facade.Count; i++)
            {
                facade[i].murder();
                facade.Remove(facade[i]);
            }
        }

        private void killButton_Click(object sender, EventArgs e)
        {
            int index_1 = projectListBox.SelectedIndex;

            openForms[index_1].Close();
            openForms.Remove(openForms[index_1]);
            facade[index_1].murder();
            facade.Remove(facade[index_1]);
            projectListBox.Items.Remove(projectListBox.SelectedItem);
        }
    }
```

The “pcmForm” class is the main form and is the longest class. The class starts off by creating two arraylists, instantiating ConcreteFactory, creating 6 strings, and creating 3 Booleans. The Strings and Booleans are modified throughout the class via the raising of certain events and serve the purpose of being passed through the chain to satisfy the arguments for the “Product” class’s constructor. The constructor for “pcmForm” only contains the “InitializeComponents()” function and names the form. The rest of the code deals with events raised via the user manipulating components on the form when the program is running. The function that is executed when the “pcButton” is clicked sets the “projectName” string to the value of the text within “projectNAmeTextBox”, adds that string as the next item within “projectListBox”, selects that item if it is the only item in the listbox, creates a new façade and simultaneously puts that façade within the “façade” arraylist, calls on the instance of “ConcreteFactory” dubbed “fmp” to execute “doSomething()”, and adds the new product to “openForms.” All of the radio button events that are coded after that manipulate the strings and Booleans created at the start of the class to contain the data needed. After all of the events for the radio buttons are finished, the click event for “resetButton” is defined. In the body of the eventhandler “projectListBox” is emptied, all of the forms within “openForms” are closed, “openForms” is emptied, and a for loop is executed that systematically kills all process via a call to the “murder()” function within the “Façade” class and empties “façade.” After the “resetButton” click event handler is the “killButton” eventhandler. In the body of this eventhandler “index_1” is created and given the value of the index of whatever item within “projectListBox” is currently selected, closes the form associated with that item, clears the, now closed, form from “openForms”, calls the “murder()” function within the instance of “Façade” that is contained in “façade” and associated with the item, removes that instance of “Façade” from “façade”, and removes that item from “projectListBox.”



Test_C

Project Name:	Test_C
Project Type:	C
IDE:	Sublime Text
Browser:	null
FTP Client:	null
Shell Interface:	null
Github Desktop Active:	False
MySQL Server Active:	False
XAMPP Active:	False

Test_Python

Project Name:	Test_Python
Project Type:	Python
IDE:	Brackets
Browser:	null
FTP Client:	null
Shell Interface:	null
Github Desktop Active:	False
MySQL Server Active:	False
XAMPP Active:	False

Test_C++_New

Project Name:	Test_C++_New
Project Type:	C++
IDE:	Visual Studio
Browser:	null
FTP Client:	null
Shell Interface:	null
Github Desktop Active:	False
MySQL Server Active:	False
XAMPP Active:	False

Programming Environments Manager

Preferences

IDE Choices
- C/C++: Eclipse, Sublime Text, [✓ Visual Studio]
- Java: [✓ Eclipse], Netbeans, Sublime Text
- Python: [✓ Brackets], Sublime Text
- Web Developement: Brackets, [✓ Eclipse], Netbeans, PHPStorm, Sublime Text, Visual Studio

Extra Software
- Browser: Chrome, Edge, Firefox, [✓ Opera], Waterfox
- FTP Client: Cyberduck, [✓ Xftp]
- Shell Interface: Command Prompt, Power Shell, [✓ Xshell]
- Github Desktop
- MySQL Server
- XAMPP

Project Types: C, [✓ C++], C#, HTML, Java, JavaScript, PHP, Python, Web

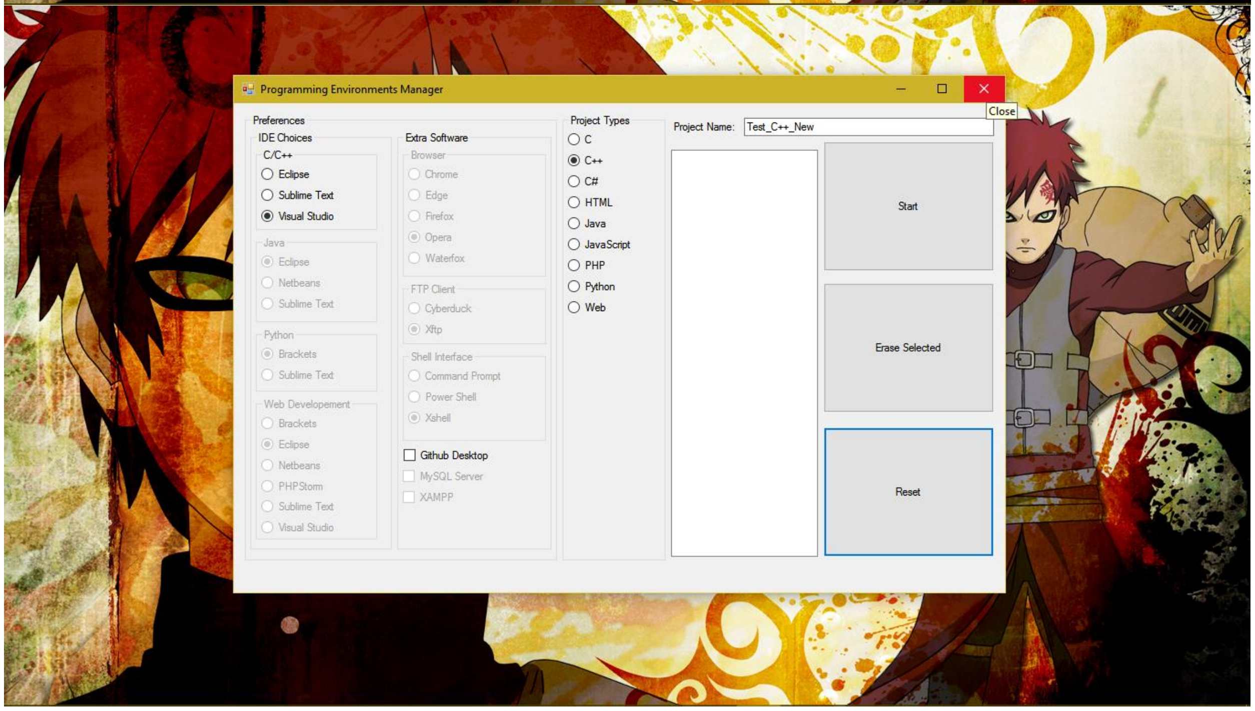
Project Name: Test_C++_New

Test_C
Test_Python
Test_C++_New

Start

Erase Selected

Reset



Programming Environments Manager

Preferences

IDE Choices
- C/C++: Eclipse, Sublime Text, [✓ Visual Studio]
- Java: [✓ Eclipse], Netbeans, Sublime Text
- Python: [✓ Brackets], Sublime Text
- Web Developement: Brackets, [✓ Eclipse], Netbeans, PHPStorm, Sublime Text, Visual Studio

Extra Software
- Browser: Chrome, Edge, Firefox, [✓ Opera], Waterfox
- FTP Client: Cyberduck, [✓ Xftp]
- Shell Interface: Command Prompt, Power Shell, [✓ Xshell]
- Github Desktop
- MySQL Server
- XAMPP

Project Types: C, [✓ C++], C#, HTML, Java, JavaScript, PHP, Python, Web

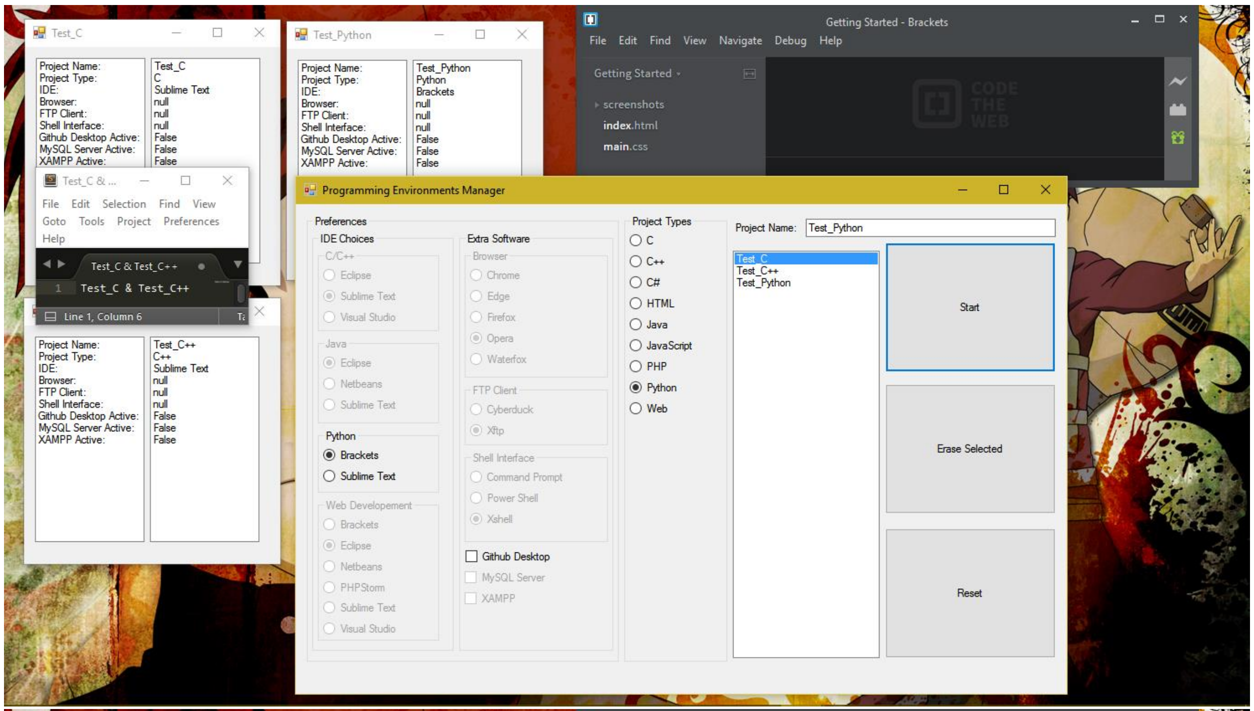
Project Name: Test_C++_New

Start

Erase Selected

Reset

Close



Project Name:	Test_C
Project Type:	C
IDE:	Sublime Text
Browser:	null
FTP Client:	null
Shell Interface:	null
Github Desktop Active:	False
MySQL Server Active:	False
XAMPP Active:	False

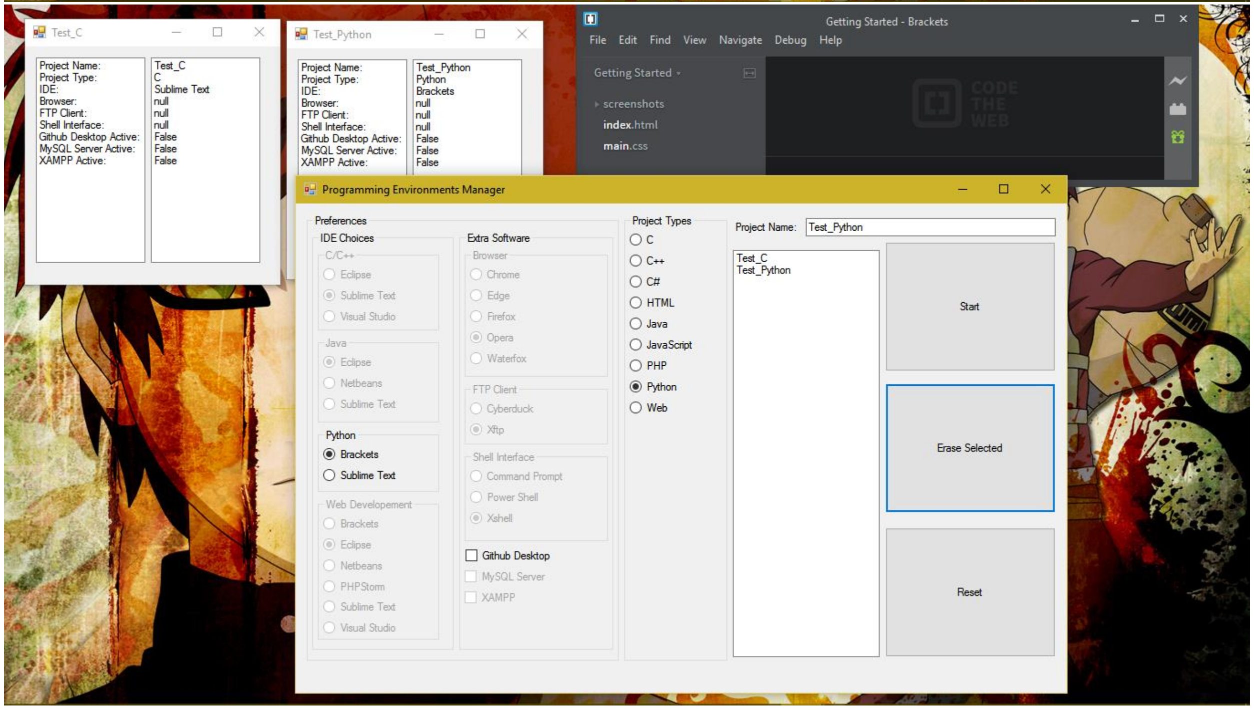
Project Name:	Test_Python
Project Type:	Python
IDE:	Brackets
Browser:	null
FTP Client:	null
Shell Interface:	null
Github Desktop Active:	False
MySQL Server Active:	False
XAMPP Active:	False

Test_C & Test_C++

Project Name:	Test_C++
Project Type:	C++
IDE:	Sublime Text
Browser:	null
FTP Client:	null
Shell Interface:	null
Github Desktop Active:	False
MySQL Server Active:	False
XAMPP Active:	False

Project Types: [✓ Python]
Python: [✓ Brackets] Sublime Text
Project Name: Test_Python

Test_C
Test_C++
Test_Python



Project Name:	Test_C
Project Type:	C
IDE:	Sublime Text
Browser:	null
FTP Client:	null
Shell Interface:	null
Github Desktop Active:	False
MySQL Server Active:	False
XAMPP Active:	False

Project Name:	Test_Python
Project Type:	Python
IDE:	Brackets
Browser:	null
FTP Client:	null
Shell Interface:	null
Github Desktop Active:	False
MySQL Server Active:	False
XAMPP Active:	False

Project Types: [✓ Python]
Python: [✓ Brackets] Sublime Text
Project Name: Test_Python

Test_C
Test_Python

The above screenshots show some of the functionality of the program.

Observations:

I enjoyed writing this program, no matter how many times it made me wanna throw my computer out. The Façade and Factory Method patterns have many practical applications which is the reason they are so commonly used.