

## How to Get Started

- Extract the zip file and use Visual Studio to open the project file “InventoryManagement.sln”.
- In Visual Studio, press F5 or the “Start” icon to start the program.
- Press the “Add” button to add parts or products. Press the “Delete” button to remove a part or product.
- When adding or modifying part, observe how the label for Company Name or Machine Number changes depending on if the user has selected the “In House” or “Outsourced” radio buttons. This also determines if the “Part” will be an Inhouse or Outsourced object. (Both Inhouse and Outsourced classes inherit from the abstract Part class)
- To search for a product or part, enter the numeric part ID or product ID in the search box and press the “Search” button.
- Press the “Modify” button to modify existing parts or products.

**Important:** Since the lookup functions defined in the requirements required an integer as an input parameter, you may only search by Product ID. Otherwise, you will get no results.

## Classes and Forms

Per Instructions, the following classes were created:

*Part.cs*

*Product.cs*

*Inhouse.cs*

*Outsourced.cs*

*Inventory.cs*

*Program.cs* was created by Visual Studio by default to house the Main() function.

The following forms were created:

*Main*

*AddPart*

*AddProduct*

*ModifyPart*

*ModifyProduct*

## Exception Controls

For Requirement J., I chose to implement the following controls:

Set 1: The “max\_textBox\_TextChanged” and “min\_textBox\_TextChanged” were added to each to the AddPart, AddProduct, ModifyPart, and ModifyProduct forms with controls with code that forces the user to enter an integer. If the user enters a blank space, a “0” will be inserted instead.

Set 2: All Delete and Cancel buttons have been coded to ask the user to confirm if he/she wants to exit.

## NOTE:

Grid views were used as lists for this project. The instructions made no mention of creating a database or spreadsheet, so all inventory items are stored in memory, in the Inventory object. A method in the Inventory class called “initializeInventory()” was created to pre-populate the inventory array lists with data to get the user started.