# Lab 1-3 Up to Speed Requirement Details

The focus of these problems will be working with information extracted from a

municipal government data feed containing bids submitted for auction of property.

The data set is provided in two comma-separated files:

1. eBid\_Monthly\_Sales.csv (larger set of 17,937 bids)
2. eBid\_Monthly\_Sales\_Dec\_2016.csv (smaller set of 179 bids)

This assignment is designed to quickly get up to speed with the C++ language and the development environment you will be using throughout this course. We will build a simple console program that uses a menu to enable testing of the logic you will complete. In this version the following menu is presented when the program is run:

| **Menu:**  **1. Enter Bid**  **2. Display Bid**  **9. Exit**  **..Enter choice:** |
| --- |

The Lab1-3.cpp program is partially completed - it contains token placeholders such as:

* ?type?
* ?variable?
* ?retval? (return value)

Replace these tokens with appropriate C++ identifiers so that the program will compile and run correctly.

Task 1: Create a new C++ Project:

1. Open the Eclipse CDT IDE previously installed (see C++ Development Installation).
2. Click the ***File*** menu in the far left corner of the menu bar and choose “***New***” then “C++ Project”.
3. In the ***Project Name:*** textbox type in “Lab1-3” with no spaces.
4. In the **Project Type** list choose “Hello World C++ Project” within the Executable section
5. In the **Toolchains** list choose the appropriate compiler for your operating system (“MinGW GCC” for Windows and “MacOSX GCC” for Mac OS).
6. Click **Finish** which will create a simple Lab1-3.cpp source file under the src directory.
   1. You’ll notice red error markers appear briefly while Eclipse finishes loading and configuring everything for the new project.
   2. At this point you can verify that everything is configured correctly by choosing “Run As -> Local C/C++ Application” and “Debug As -> Local C/C++ Application”
7. Download the starter program source (Lab1-3.cpp) and copy it to the src directory, replacing the existing auto-generated one. Remember to right-click on the project in the Project Explorer pane on the left and **'Refresh'** the project so it adds all the new files to the src folder underneath.

Task 2: Define a data structure to hold bid information together as a single unit of storage. Elements include:

1. the title of the bid item;
2. the fund the proceeds will be credited to;
3. the vehicle Id; and
4. the amount of the bid

Task 3: Declare instance of data structure to hold bid information.

1. Within the main() method you will need to declare a variable that is an instance of the data structure type you defined in Task 1.

Task 4: Store input values in data structure.

1. Within the getBid() method you will need to complete the prompt statements by filling in the tokens with appropriate variables as well as populate the return value of the method.

Task 5: Display the bid values passed in data structure.

1. Within the displayBid() method you will need to complete the output statements to display the bid information passed in to the method.

Task 6: Complete the method calls then test the program.

1. Within the main() method complete the method calls in the switch statement then test the program by running it.

Here is sample output from running the completed program to illustrate the separate activities:

| **Example Input** | **Choice: 1** | **Choice: 2** | **Choice: 9** |
| --- | --- | --- | --- |
| **Display** | Menu:  1. Enter Bid  2. Display Bid  9. Exit  Enter choice: 1 | Menu:  1. Enter Bid  2. Display Bid  9. Exit  Enter choice: 2 | Menu:  1. Enter Bid  2. Display Bid  9. Exit  Enter choice: 9 |
| **Output** | Enter title: Hoover Steam Vac  Enter fund: GENERAL  Enter vehicle:  Enter amount: $27.00 | Title: Hoover Steam Vac  Fund: GENERAL  Vehicle:  Bid Amount: 27 | Good bye. |