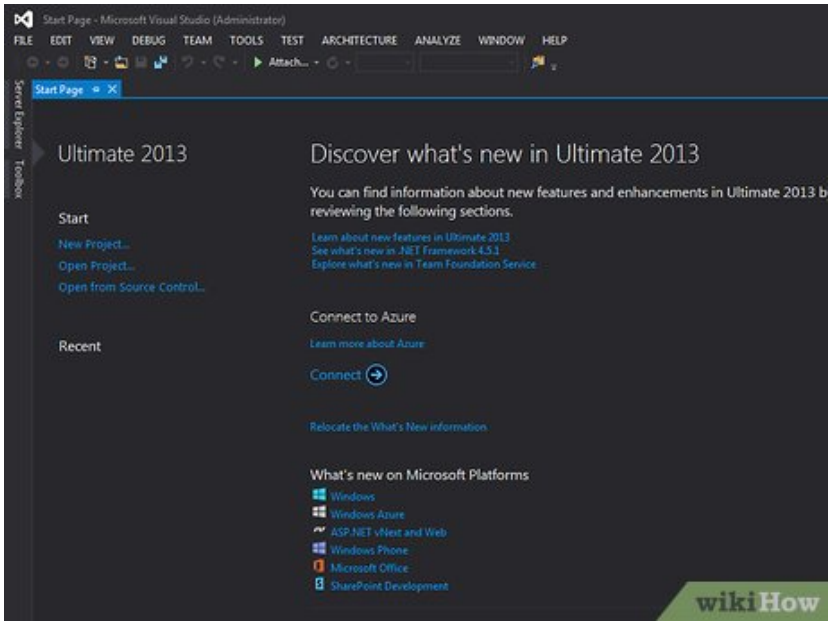
How to Allocate Dynamic Memory (C++)

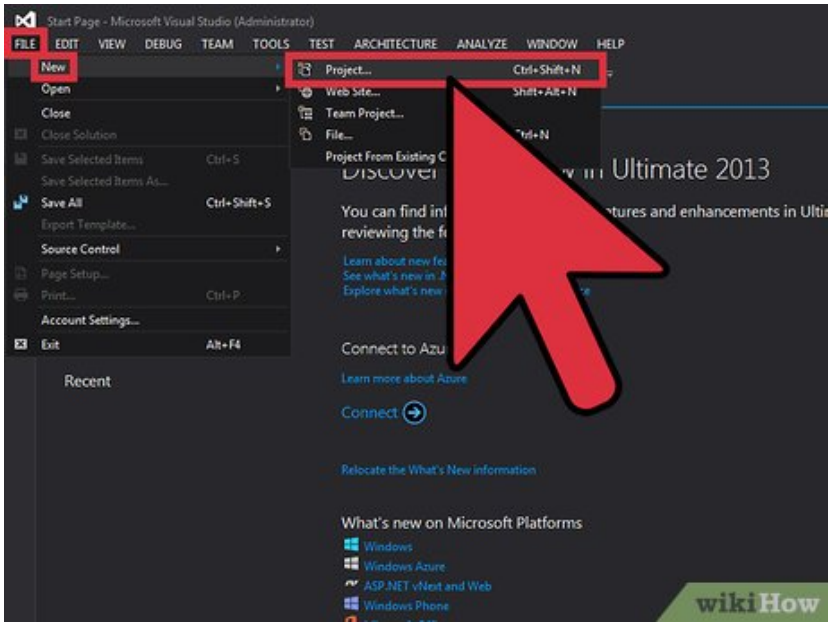
For most beginning programmers, the memory needs for the programs they have written have been predetermined before their program runs, such as when declaring the size of an array. But sometimes, we as programmers do not know exactly how...

Part 1 of 2:

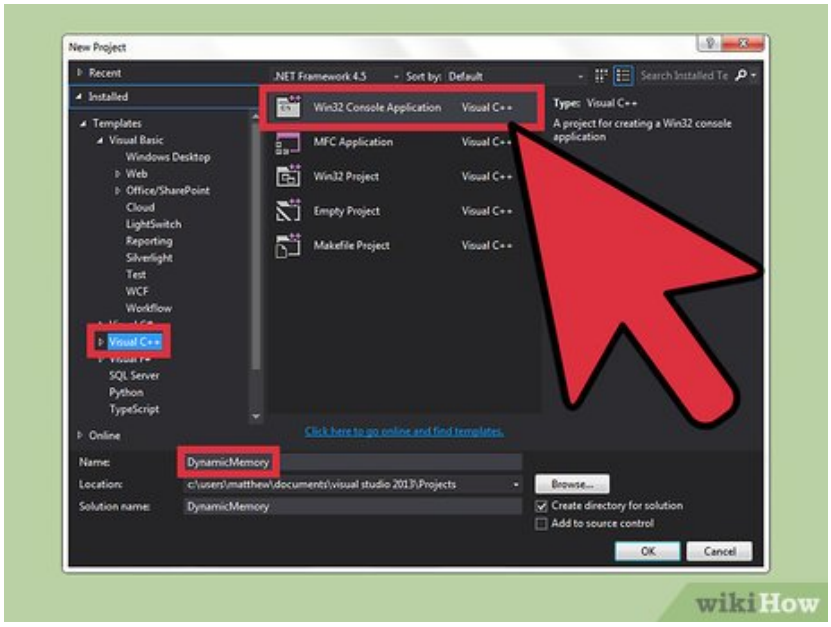
Getting Started

1. 

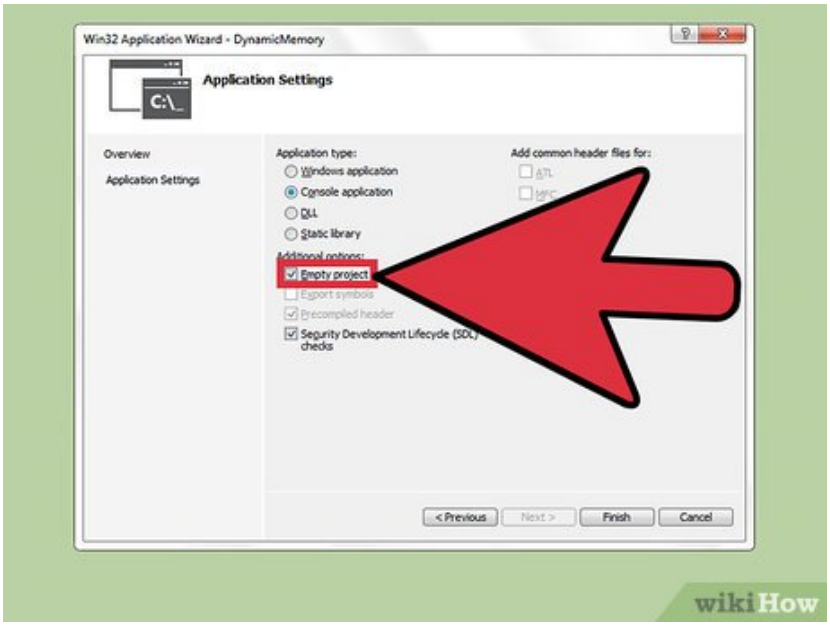
Open Visual Studio 2013 (or your choice of compiler). Your start-up screen should look similar to this.

2. 

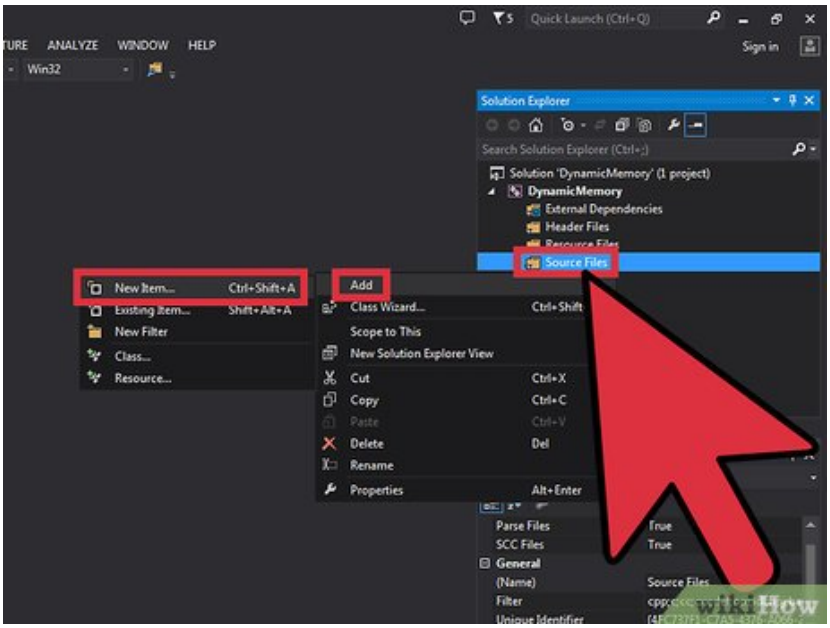
Create a new project. Click "File" and then "New Project".

3. 

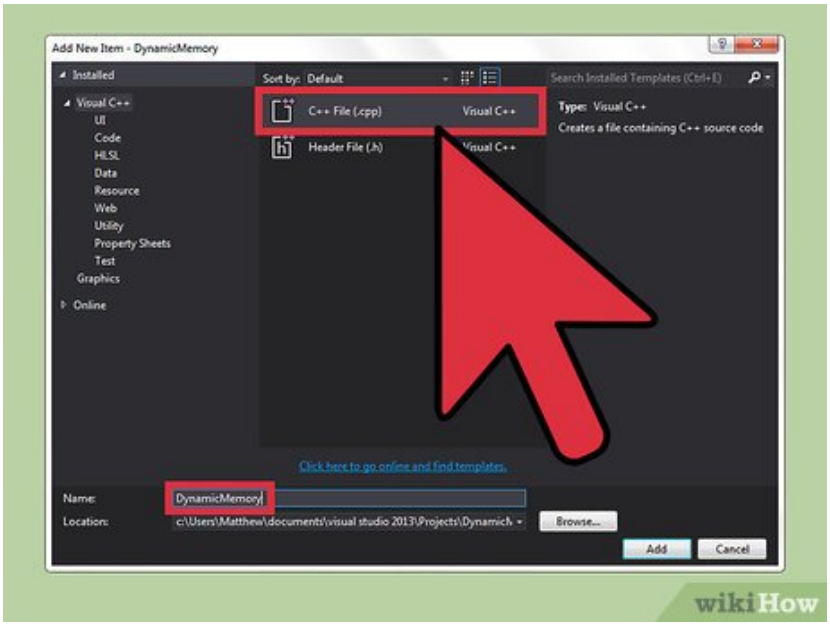
Select "Win32 Console Application" and name your project "DynamicMemory".

4. 

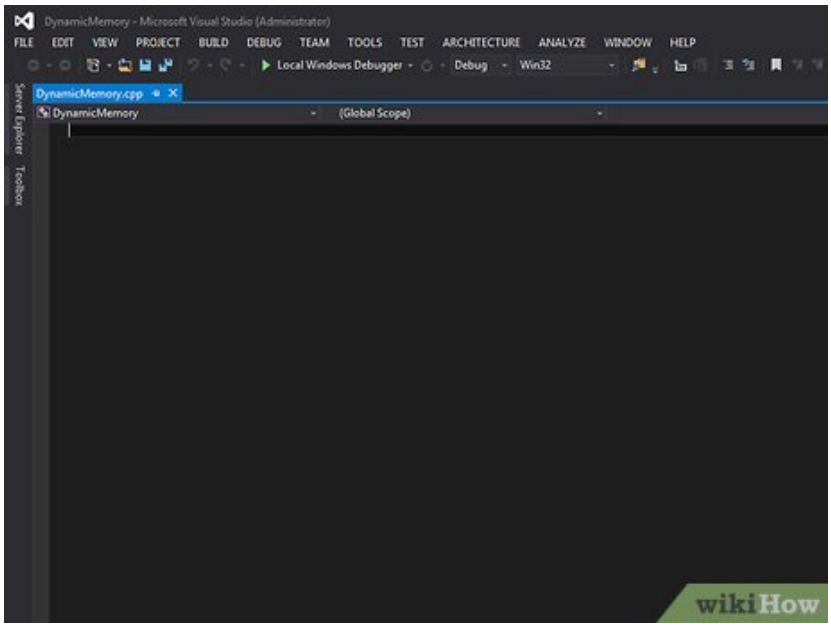
Check "Empty project" and click "Finish" to create your project.

5. 

Right click on "Source Files" on the right side of your screen. Then click "Add" followed by "New Item". (You can also accomplish this by hitting Ctrl +? Shift + A on your keyboard).

6. 

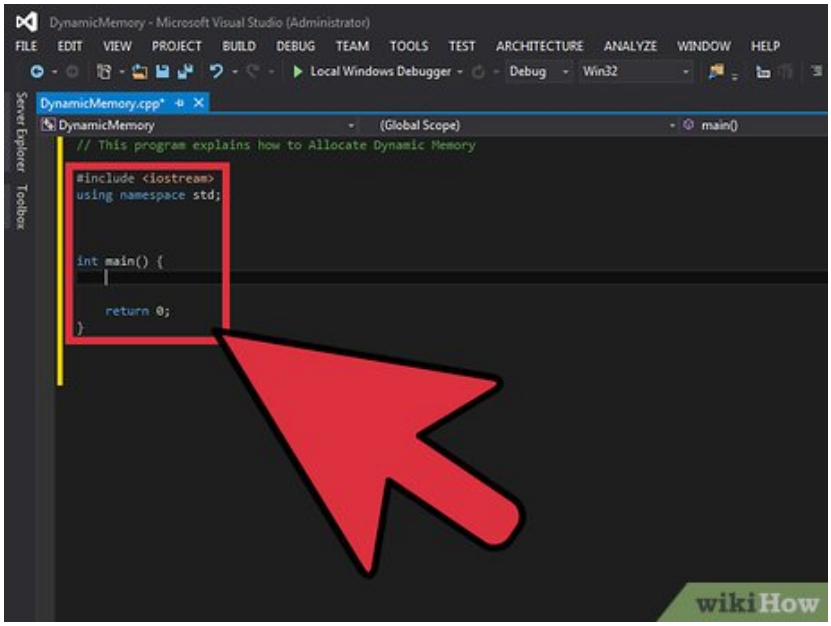
Select "C++ File (.cpp)". Name the file "DynamicMemory".

7. 

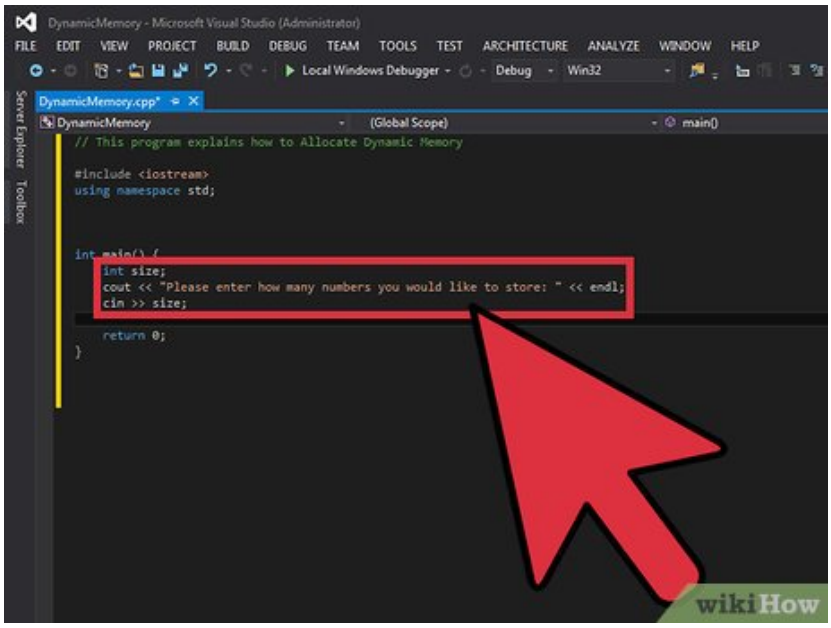
You now have a new project and are ready to begin coding.

Part 2 of 2:

Allocating Dynamic Memory

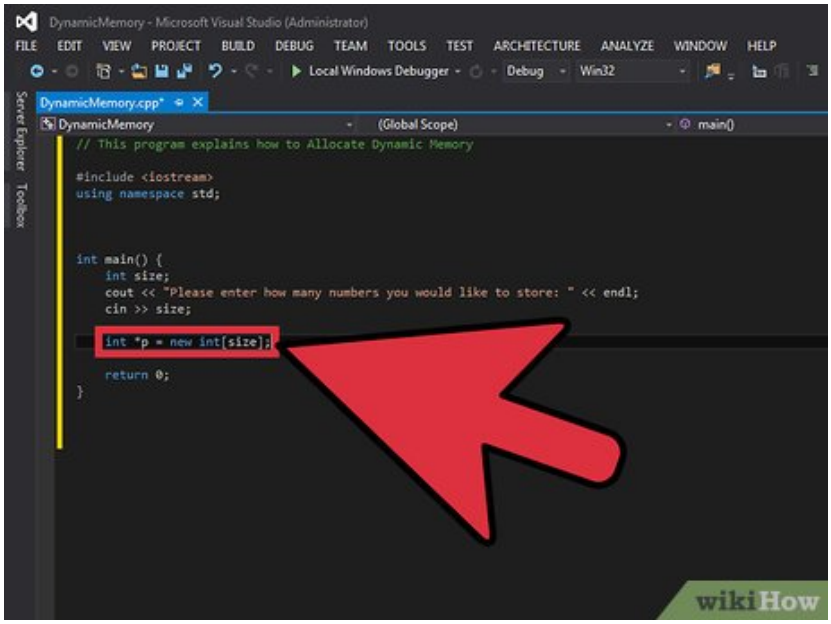
1. 

Begin by including the standard iostream header and main function.[1]

2. 

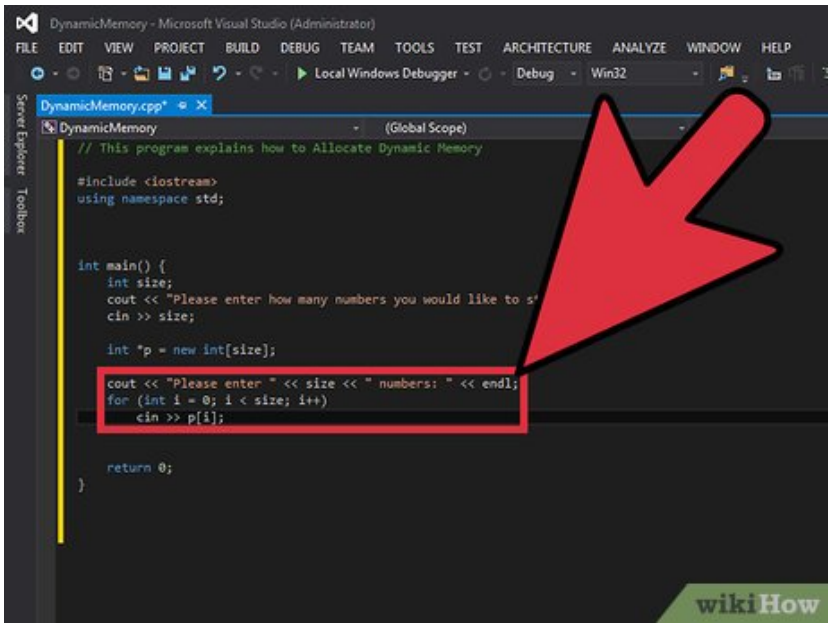
Prompt the user.

1. Declare a variable called "size" of type int.
2. Use cout to prompt the user to input how many numbers they would like to store.
3. Use cin to store the value they entered into the variable "size".

3. 

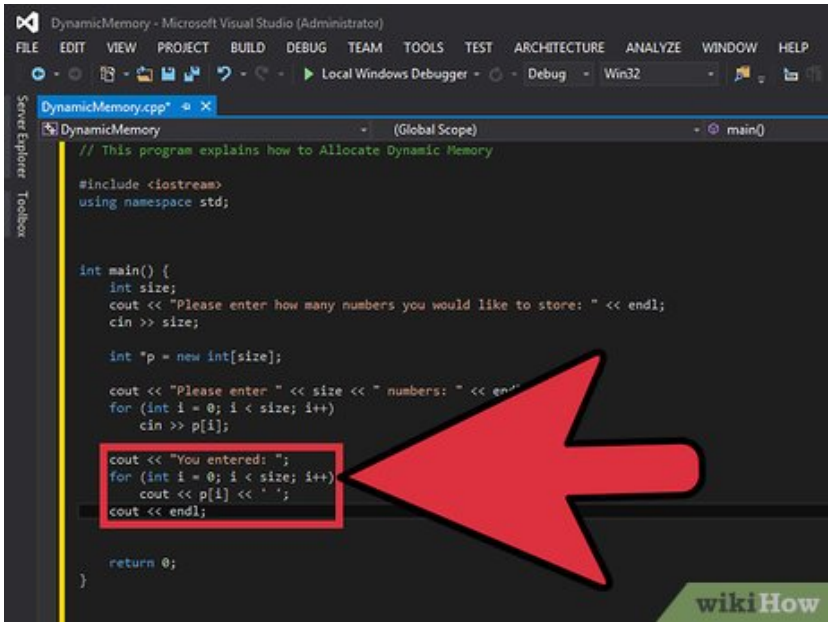
Declare a Dynamic Variable.

1. Dynamic memory is allocated by using the operator "new"
2. "new" allocates memory as a variable (in this case of type int) and returns a pointer to it that we have stored in "p"
3. The "size" variable is then used to dynamically create an array of the exact size that the user has input.

4. 

Prompt the user to enter numbers.

1. Use cout and the "size" variable to prompt the user to input the requested number of numbers.
2. Use cin and a for loop to store the user input into the dynamic array.

5. 

Output results.

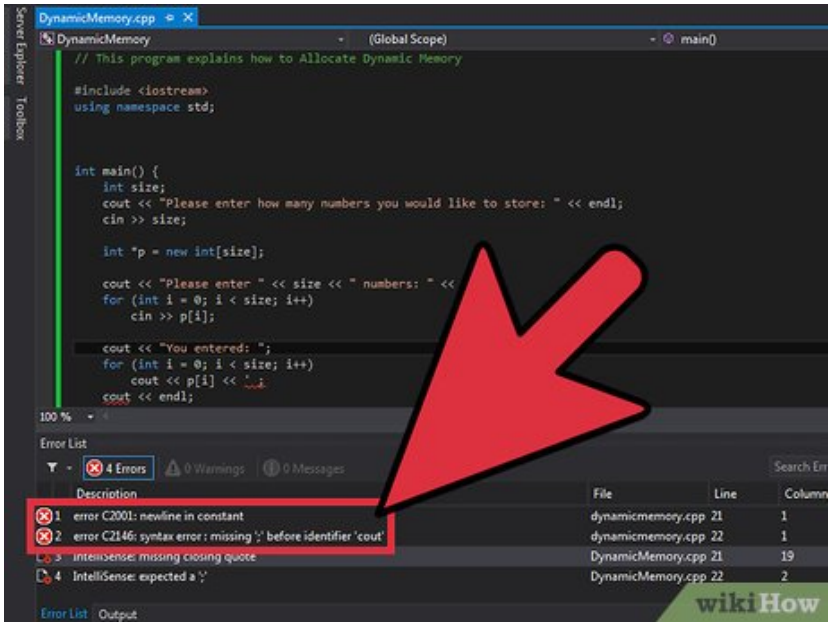
1. Use a cout statement and a for loop to output the contents of your dynamic array.

6. 

Run your program.

1. Enter Ctrl + F5 on your keyboard to run your program.
2. Test your program to verify proper execution.

7. **Deallocate memory.** Every piece of memory allocated with "new" should be deallocated using the "delete" operator.
    1. When deallocating C-arrays, remember to use brackets: "delete [] ".

8. 

Troubleshoot if needed. If you attempt to run your program and you receive a build error, check the bottom of your screen and Visual Studio can generally advise you to the nature of your error.

1. The most common issue if a program will not compile is a syntax error. Be sure to check all your statements end with the proper semicolon ";".
2. Visual Studio will highlight most syntax errors in red to help you locate them
3. In the event your program will not compile and no syntax errors are found, try Cleaning your solution by clicking "Build" and then "Clean solution". This removes build artifacts from your previous build.