



HARRIS STUDENT DESIGN CHALLENGE, 2019
PROFESSOR AARDVARK'S ASTEROID JOURNEY
HELP / TIPS / HINTS / EXAMPLES

VERSION 1.4

HARRIS®

Visit often as content is updated frequently

[Mac OS Formation App Installation](#)

[Mac OS Run Formation App](#)

[Mac OS Install Python Nav API](#)

[Windows Install Python](#)

[Windows Formation App Installation](#)

[Windows Run Formation App](#)

[Windows Install Python Nav API](#)

[Windows Install C++ Compiler](#)

[Windows Install Eclipse IDE](#)

[Windows Install Java – Video Tutorial](#)

[Windows Install Eclipse IDE – Video Tutorial](#)

[Windows Formation App Observations](#)

[Linux Install G++ \(C++ Compiler\)](#)

[Windows Install Visual Studios IDE](#)

[Python Mac/Linux/Windows Virtual Environments Videos](#)

[Windows: Installing pyzmq in a Python Virtual Environment](#)

[Windows: Python API Test – Tips](#)

- Visit www.Codevark.org
- Click on Downloads
- Click on Mac OS installation button
- A zip file, “formation_osx64.zip”, will be placed in your “Downloads” folder
- Double-click “formation_osx64.zip”; folder “formation_osx64” will appear
- Move the folder “formation_osx64” to your Documents folder
- Double-click to open folder “formation_osx64”
- You should see two .json files and executable “formation.app”
 - This executable is not signed and you must override your security settings to open it
- To run, press the Control key and click the app icon, then choose Open from the shortcut menu
- Confirm by clicking Open
- Subsequent double-clicks on the app icon should run the app
 - If not, try Control-click, Open again

- **Use Finder to go to folder “formation_osx64”**
- **First time**
 - To run, press the Control key and click the app icon, then choose Open from the shortcut menu
 - Confirm by clicking Open
- **Subsequent times**
 - Double-click the app icon
- **The Formation Configuration dialog box allows you to set parameters for running the app**
 - Screen resolution: Use default or select a smaller screen size if you don't want full-screen
 - Graphics Quality: Select what you think your computer can handle; adjust as needed
 - Windowed: Check box is you want a window you can minimize and move around
- **Press Play**

- **Pick a Challenge from the drop-down box**
 - We will add or publish more of these as we can
- **Choose how to play**
 - Play And Host: Make your computer a Host that you and others can play on
 - Join A Development Server: Enter the IP address of a computer that chose the Play And Host option above
 - Web Server: When we get the full Codevark Web Server running, this option will be enabled
- **Wait for others to connect and/or click Join to start the Challenge**
- **If prompted, Allow accepting incoming network connections**
- **The formation with your drone ship should appear**
- **Use up, down, left, right arrow keys to make short thruster bursts and move your ship**
- **Use 's' to toggle score view**
- **Use 'v' to toggle View Options view; adjust view options as you desire**
- **Use 'Space' to apply Emergency Stop**
- **Use 'Esc' to exit**

[Back to TOC](#)

- Visit www.Codevark.org
- Click on Downloads
- Click on Python Nav API installation button
- A zip file, “aardmsg_python.zip”, will be placed in your “Downloads” folder
- Double-click “aardmsg_python.zip”; folder “aardmsg_python” will appear
- ~~Move the folder “aardmsg_python” to your Documents folder~~
- ~~Double-click to open folder “aardmsg_python”~~
- ...To be continued...

- **Download Python 3.x for Windows from python.org**
- **Install Python (recommended installation in order to make sure you get Idle)**
- **Open Idle which starts with a Python Shell to run your program**
- **Use File->New File to start a new program**
- **In the new window that opens, write your program**

```
print("Hello, World!")  
print("This is my first Python program!")
```
- **To run your program, choose Run->Run Module from the window that you are typing your program in.**
- **Idle will prompt you to save it if you haven't already**
- **Your program will execute in the Python Shell window that started up initially**

- Visit www.Codevark.org
- Click on Downloads
- Click on Windows installation button
- A zip file, “formation_win64.zip”, will be placed in your “Downloads” folder
- Double-click “formation_win64.zip”; folder “formation_win64” will appear
- Move the folder “formation_win64” to your Documents folder
- Double-click to un-zip the file contents
 - Comment: I experienced a problem when trying to un-zip this file. I installed 7-zip from 7zip.org and changed my computer's settings for .zip files from using VLC to using 7-zip.
 - After installing 7-zip, a right-click on formation_win64 has a 7-zip menu option that lets you extract the file.
- Extracted the formation_win64 folder and its contents

- **Double-click the app icon**
- **The Formation Configuration dialog box allows you to set parameters for running the app**
 - Screen resolution: Use default or select a smaller screen size if you don't want full-screen
 - Graphics Quality: Select what you think your computer can handle; adjust as needed
 - Windowed: Check box is you want a window you can minimize and move around
- **Press Play**
- **Pick a Challenge from the drop-down box**
 - We will add or publish more of these as we can
- **Choose how to play**
 - Play And Host: Make your computer a Host that you and others can play on
 - Join A Development Server: Enter the IP address of a computer that chose the Play And Host option above
 - Web Server: When we get the full Codevark Web Server running, this option will be enabled

- **Wait for others to connect and/or click Join to start the Challenge**
- **If prompted, Allow accepting incoming network connections**
- **The formation with your drone ship should appear**
- **Use up, down, left, right arrow keys to make short thruster bursts and move your ship**
- **Use 's' to toggle score view**
- **Use 'v' to toggle View Options view; adjust view options as you desire**
- **Use 'Space' to apply Emergency Stop**
- **Use 'Esc' to exit**

- Downloaded aardmsg_python and extracted it
- ...To be continued...

This is a step-by-step walkthrough on how to install the C++ compiler mingw-w64 to build and run any C++ code. Mingw-w64 is an improved version of MinGW that supports both 32 and 64 bit.

Installation steps:

1. Download and run the installer found at <https://mingw-w64.org/doku.php/download/mingw-builds>
2. Settings for installation:
 - Latest version (8.1.0)
 - Architecture: x86_64 if you are running a 64 bit system.
Check by typing windows key + system information.
 - Threads: posix to enable C++11 <thread>, <mutex> and <future>
 - Exception: sjlj as supported by gcc
3. Install somewhere easy to find. I used "C:\mingw64".
4. Wait for it to install (5 minutes)

Setting the Path environment variable so that your command prompt can compile and make from anywhere:

First, go to where you installed mingw. Since I installed it in C:\, I navigate to C:\mingw64.

Inside your install directory, open the mingw64 folder. Inside that you should see a "bin" folder.

This contains all of the executables we need to compile C++ programs. We need to add that to the Path variable so that the command prompt will know where to go to compile programs.

- 1. Type windows key + "path"**
- 2. Click "Edit the system environment variables"**
- 3. In the advanced tab, at the bottom, click "Environment Variables..."**
- 4. In the bottom half "System Variables", find and click "Path" then "Edit"**
- 5. Click "New" to add a new variable.**
- 6. This is where we add the bin folder. If your installation is in "C:\mingw64" add "C:\mingw64\mingw64\bin" without the quotes. If you installed somewhere else, then you would type "<Install_Path>\mingw64\bin".**

Finally we change the name of the make executable so we can run that by simply typing "make".

1. Navigate to your bin folder (the same one you added to the Path variable).

Inside that you should find a file named "mingw32-make.exe".

2. Rename it to "make.exe".

To confirm that the compiler is working, open a command prompt and type "gcc --version" and "make --version".

You should see output describing the version installed.

If you see "... is not recognized as an internal or external command, operable program or batch file."

then something went wrong with your installation or you did not add the correct path to your bin folder to the environment variables.

Now you should be able to compile a C++ program by using "g++ programname.cpp".

This is a step-by-step walkthrough on how to install Eclipse to write, build, and run C++ code using mingw-w64. If you are using any other compiler, you may have to link it at the appropriate steps.

Installing Eclipse:

1. **Go to <https://www.eclipse.org/downloads/> and click the "Download 64 bit" button under "Get Eclipse IDE 2019-03".**
Note that the current Eclipse version is 64 bit only. If you are on a 32 bit system you will either need an old version, or a different environment.
2. **Run the installer.**
3. **Click "Eclipse IDE for C/C++ Developers".**
4. **Pick an installation folder. I used "C:\Users\MyUsername\eclipse\cpp-2019-03".**
5. **Accept licenses to continue.**
6. **Launch when done installing.**

Before Setting up Eclipse (USING MINGW64):

Since Eclipse looks for certain executables to do indexing we must ensure that our mingw has the proper executable.

1. Navigate to your mingw64 bin. For example, mine is C:\mingw42\mingw64\bin.
2. You should find a file named "x86_64-w64-mingw32-gcc-8.1.0.exe".
3. Copy and paste it into the same bin folder. There should now be a "x86_64-w64-mingw32-gcc-8.1.0 (1).exe".
4. Rename the copy to "mingw-32-gcc.exe". Make sure it is exactly that name or Eclipse won't be able to index!

Setting up Eclipse:

1. The first time launching eclipse you must select a workspace. I use the default one.
2. Select "Create a new C/C++ project".
3. Select "C++ Managed Build".
4. Give your project a name, select "Hello World C++ Project" (Or empty C++ project), and select the Toolchain "MinGW GCC".

If MinGW GCC doesn't appear for you, make sure that you renamed the executable in the bin folder described in the previous section. Otherwise, try unchecking "Show project types..." at the bottom.

5. Click finish to open up an example C++ program.

[Back to TOC](#)

6. If everything succeeded, symbols such as "std" and "cout" should not be underlined.
7. Type "Ctrl+b" to build the program. The console at the bottom should show that it compiled.
8. Type "Ctrl+11" to run, select local application to run in eclipse.
9. If the console displays "!!!Hello World!!!" then congratulations, you successfully set up your C++ compiler and Eclipse!

On the left hand window, you can see your project explorer, showing your projects files.

On the right is the outline, which displays classes, variables, and more.

On the bottom is the console.

You can minimize and close these as desired. If you want to get one back, go to Window->Show View-> and open the desired view.

One of the most powerful tools of eclipse is autocomplete. When typing, if you press "Ctrl+Space" a list of completions will show.

(Note: If you chose the Hello World C++ Project, autocomplete might not work in the initial generated cpp file, but should work in all new files you create.)

You may want to change the appearance of eclipse, such as the theme.

To change the theme (Light, Dark, etc...):

Window->Preferences->General->Appearance. Under the Theme dropdown, select a theme (such as dark) and click "Apply and Close".

Happy coding!

- **Installing Java on a Windows 10 laptop**
- **From one of our volunteers:**

I installed Java JDK 8 and followed the following tutorial:

<https://www.youtube.com/watch?v=rzto4yY3pVw>

- **Success!**
- **Continue with [Windows Install Eclipse IDE – Video Tutorial](#)**

[Back to TOC](#)

- **Installing Eclipse on a Windows 10 laptop**
- **From one of our volunteers:**

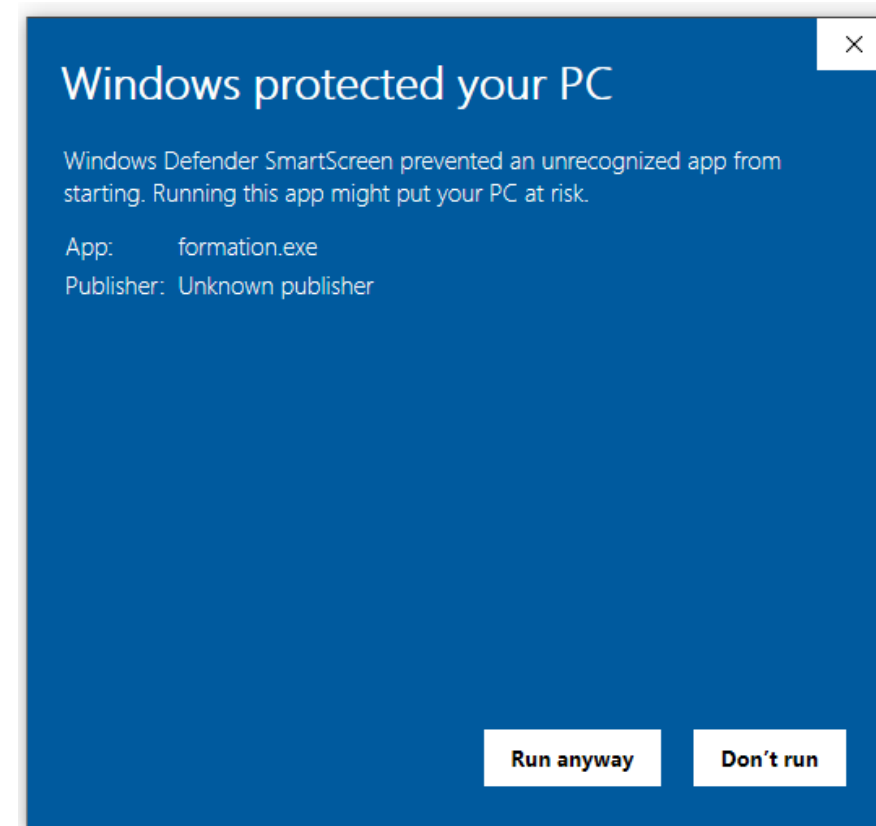
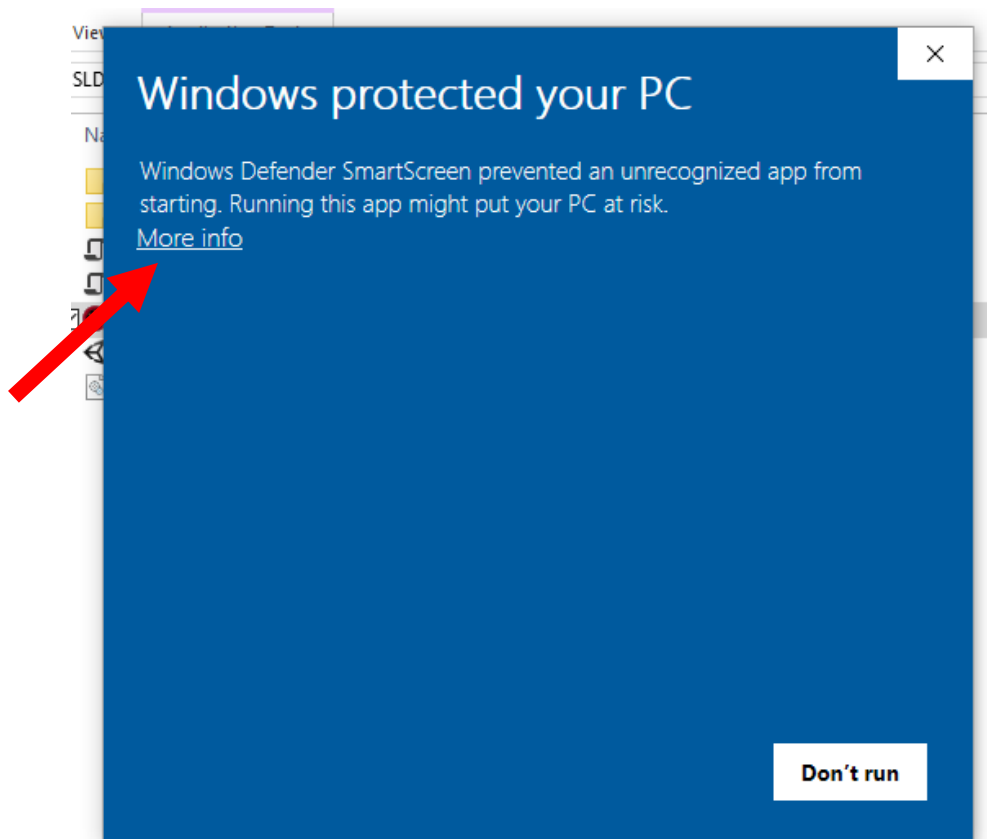
I followed the following tutorial for installing Eclipse:

<https://www.youtube.com/watch?v=CHb6HBo2h7U>

- **Success!**

[Back to TOC](#)

- See elsewhere for installing the Formation Application
- Observations:
 - Windows shows a warning when trying to run formation.exe; to get it to run the user needs to click on more info then “Run anyway”



[Back to TOC](#)

This is a step-by-step walkthrough on how to install the linux C++ compiler, G++.

Check if Installed:

- **Open a terminal and run the command: `g++ -v`**

If you get a version number back, you already have g++ installed.

If not follow these Installation steps:

1. **In your terminal, enter the command: `sudo apt-get update`**
2. **In your terminal, enter the command: `sudo apt-get install g++`**

This should install g++ for you. In order to check if the installation was successful, try re-running the `g++ -v` command on the terminal.

Creating the Hello World App:

1. Create a new text file in a location of your choosing. Name that text file, “Main.cpp”.
2. Inside of that text file, enter the following code:

```
#include <iostream>

int main(void){
    std::cout << “Hello World\n”;
    std::cin.get()
    return 0;
}
```

1. Save Main.cpp.
2. Open a terminal in the same directory as your Main.cpp file.
3. Enter the following command: `g++ -o hello Main.cpp`
4. This will compile your Main.cpp file into an executable named *hello*.
5. Run this by entering the following command into the same terminal: `./hello`

You should see the terminal display “Hello World”! To exit the program, just hit enter.

[Back to TOC](#)

This is a step-by-step walkthrough on how to install Visual Studios to write, build, and run C++ code.

Installing Eclipse:

Go to <https://visualstudio.microsoft.com/downloads/> and click the "Free download" button under "Community".

- 1. Run the installer.**
- 2. When asked what type of development you will be using Visual Studios for, Click "Desktop development with C++".**
- 3. Launch when done installing.**

Creating a Hello World App:

- 1) With Visual Studio running, click through **File→New→Project...**
- 2) In the New Project window that pops-up expand **Visual C++**, and click **General**.
- 3) You should now see a list of C++ projects you can create. We will select the option for **Empty Project**. Once selected, change the **Location** field towards the bottom to point to wherever you want the application to be saved to.
- 4) Change the **Name** field to “**Hello World**”.
- 5) Click “**OK**”. Visual Studio will now create an empty C++ project for you.

Setting up the Hello World App:

You should see a window for Solution Explorer on either the right or left of the Visual Studio console. Inside this window is the layout of your application. For right now, we mostly care about the Source Files folder.

Right click on the Source Files folder and click Add->New Item...

- 1. This should bring up the Add New Item window. In it, you should see a list of C++ file types you can create. We will select the C++ File (.cpp) option. In the Name field, change the value to “Main.cpp”, and finally click Add.**

Visual Studio will then create a new file, Main.cpp. This file will be automatically opened in the Visual Studio console and allow you to start writing the code for your application.

Now, enter the following lines of code into the “Main.cpp” file.

```
#include <iostream>

int main() {
    std::cout << “Hello World” << std::endl;
    std::cin.get();
    return 0;
}
```

This code will print “Hello World to the command line!

Now, you are finally ready to run your application. Hit the Run button towards the top of the console (Big green right facing arrow). A console should appear, with your “Hello World” text displayed. To exit, just hit enter.

The following YouTube links will take you to a favorite Python development channel

Follow the link for your OS for help setting up Python Virtual Environments

[Mac / Linux](#)

[Windows](#)

[Back to TOC](#)

On Windows, installing pyzmq using a Python virtualenv:

Open a Command Prompt window. The commands below are to be entered in that Command Prompt window

Use the path where your Python installation is. I also saved my programs there.

```
>cd C:\Users\clara\AppData\Local\Programs\Python\Python37-32
```

Add a virtual env

```
>python -m venv env
```

Activate the virtual env

```
>env\Scripts\activate
```

Install zmq in the virtual env

```
(env) >pip install pyzmq # Note that the -upgrade option was not available
```

My program still couldn't find zmq when I ran it so I used this to add it to the search path.

```
(env) >set PYTHONPATH=C:\Users\clara\AppData\Local\Programs\Python\Python37-32\env\Lib\site-packages
```

[Back to TOC](#)

Below are tips for writing your first Python program using the aardmsg_python API:

Moved aardmsg_python to same directory where my Python program is so it could find it.

Updated import in my test program to specify 'from aardmsg_python.aardmsg import AardMsg'.

You could also add the path to your PYTHONPATH environment variable instead.

IMPORTANT: Start Formation server first by double-clicking on formation in whichever folder you unzipped it in.

Select the checkbox next to Windowed. You may need to adjust your resolution as well.

Click the Play! button

Click the Play and Host button

Click Join

Run your program from the virtualenv in the Command Prompt window:

```
(env) C:\Users\clara\AppData\Local\Programs\Python\Python37-32>python AardMsg_API_Test.py
```

[Back to TOC](#)