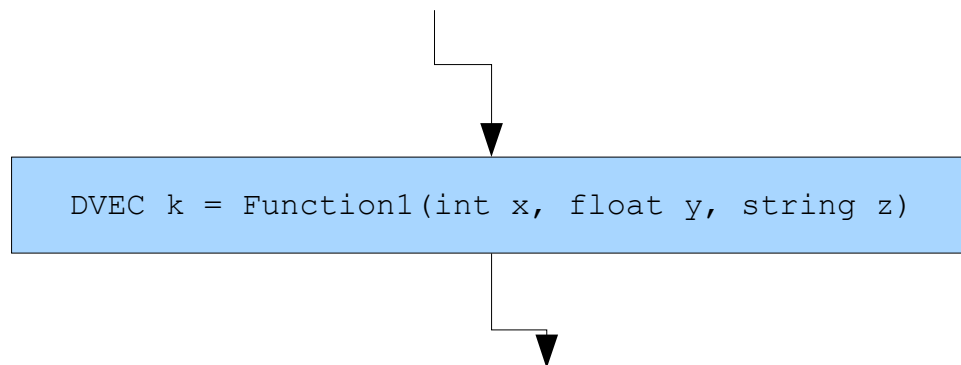
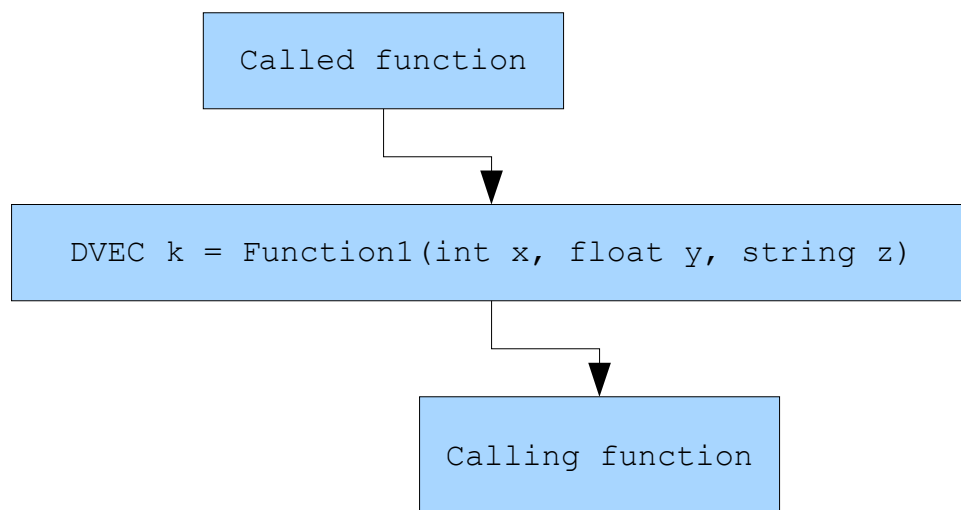


1. Function boxes shall contain the prototype of the function, with the input and output variables' datatypes declared. For example:



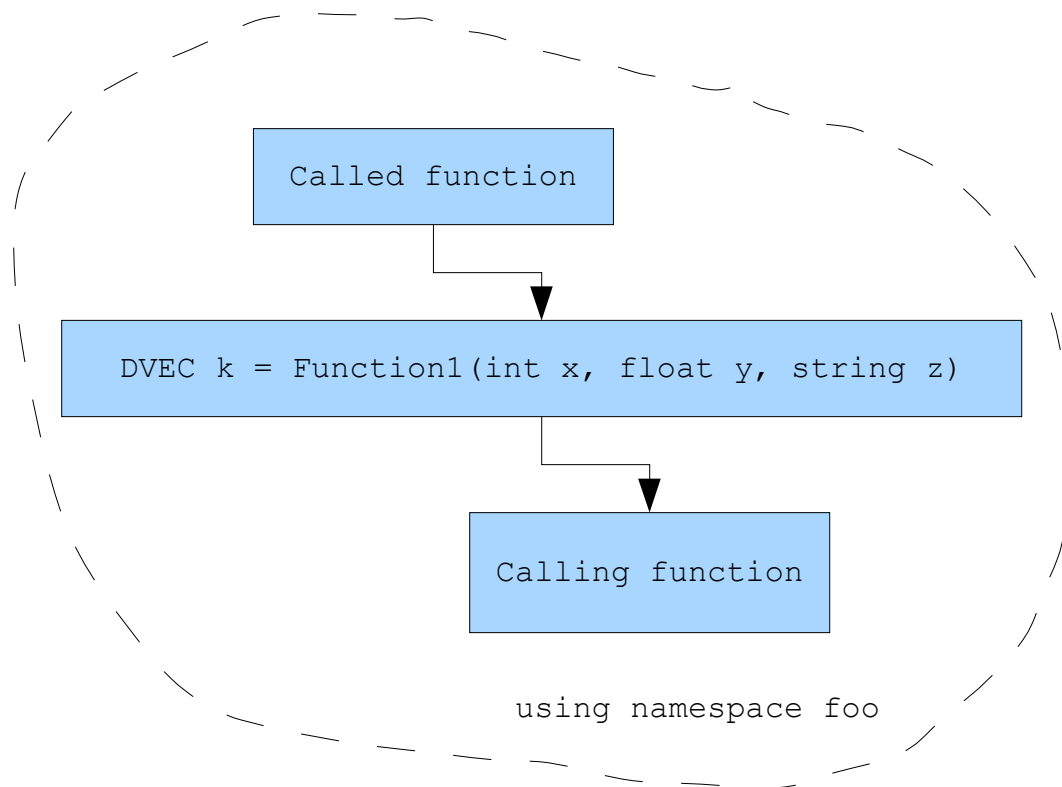
Users will put the datatypes that correspond to their language, e.g. dict in Python.

2. Arrows feeding into the function box mean that that data is being put into/used by that function. Arrows leaving that function box mean that data is being passed/called to another function (or is being output). For example:

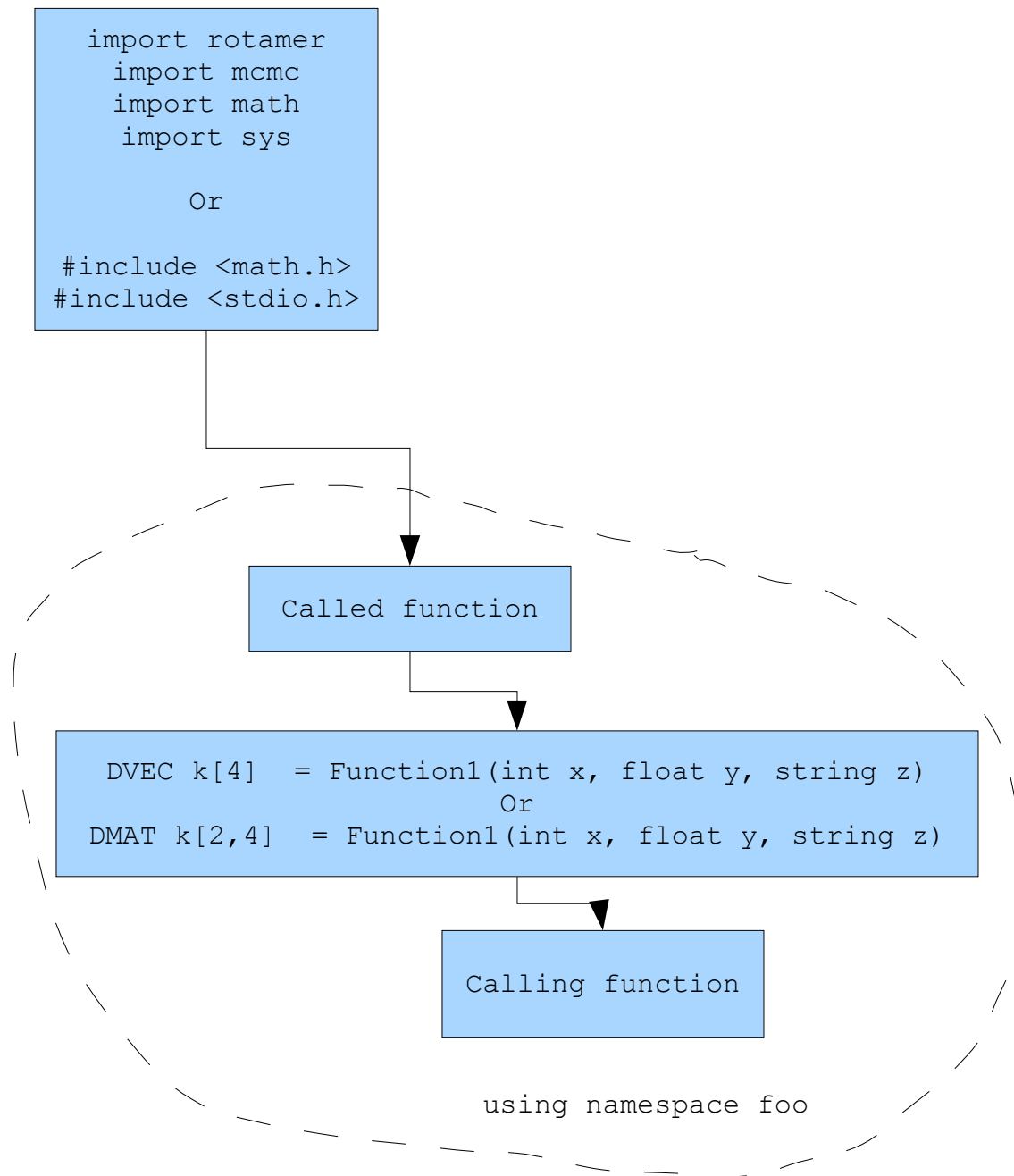


3. The general form of these code documentation diagrams will be flowcharts, illustrating the actual execution of the code. The used shapes will be boxes for functions, diamonds for flow control code (e.g. if, for, while, etc.). For `ifdef` and `ifndef` in C/C++, we will just use diamonds. For namespaces, we will use the curve tool to draw a dotted curve around the boxes to which

that namespace is allotted. We use the curve because the namespaces could apply to strange topologies on the flowchart. For example:



4. For C/C++, #includes, #defines, and imports in Python that are declared at the beginning of the code will be put in a big box at the start of the function. For example:



5. Again, the only shapes to be used will be boxes and diamonds.
6. For multidimensional datatypes, like NumPy matrices, NumPy arrays, or DMAT, DVEC, the dimensions will be denoted by a square-bracketed ordered pair, denoting rows and columns. This will be placed in the prototype in the function box. For example:

```
DVEC k[4]  = Function1(int x, float y, string z)
Or
DMAT k[2,4] = Function1(int x, float y, string z)
```

7. In the upper right hand corner, a hyperlink will be added to the original code file.
8. In the import/include box, hyperlinks will be added to the source code we have created, and important 3rd-party add-ons, but not in-built modules/header files. For example, one would hyperlink to `<ublastypes.h>` because it is an important 3rd-party package we need for our code, or to `<MCMC_AS_9-6.h>`, because it is our own code.
9. Written comments, like what Karthik wrote in his handwritten protein code documentation PDF file, will be typed up separately in a OpenOffice file, and then hyperlinked to in the document, in the upper-right hand corner, below where the source file hyperlink is.
10. For the color scheme, we will use just the default OpenOffice blue color, **blue 8**. The font will be `Courier New`.
11. For C++ header files, the prototypes and functions defined therein shall be treated the same way as described above, with each function/prototype getting a separate box. For example:

```
DVEC[4]= function1(int x)
```

```
int k = function2(string x)
```

```
double k = function3(double  
x, double y)
```

```
...
```

12. In drawing the protein schematics, all angles will be **bold light blue Courier New**.
13. Arrows showing the progression of the code, e.g. step by step, will be **bold light blue**.
14. Information taken from external files will be given a **light green box**, and connected to the calling function by a regular black arrow.
15. Pointed values that are edited by a function will be placed in a light blue box, and then pointed at with a fine-dotted-line arrow; those called by a function will have the arrow emanating from it instead.