**Pybind special cases**

Handling virtual function calls and inheritance: [Overriding virtual functions in Python](http://pybind11.readthedocs.io/en/master/advanced/classes.html?highlight=virtual%20function#overriding-virtual-functions-in-python)

Adding default arguments to function calls: [Default arguments](http://pybind11.readthedocs.io/en/master/basics.html#default-arguments)

Getter and setter functions in python: [Return value policies](http://pybind11.readthedocs.io/en/master/advanced/functions.html?highlight=getter#return-value-policies)

Static functions: [Creating bindings for a custom type under See also](http://pybind11.readthedocs.io/en/master/classes.html?highlight=static#creating-bindings-for-a-custom-type)

Handling enum classes: [Enumerations and internal types](http://pybind11.readthedocs.io/en/master/classes.html?highlight=enum#enumerations-and-internal-types)

std::function in the argument: [Return value policies](http://pybind11.readthedocs.io/en/master/advanced/functions.html?highlight=getter#return-value-policies)

Overloading functions: [Overloaded methods](http://pybind11.readthedocs.io/en/master/classes.html#overloaded-methods)

Documentation used for python shell:

<https://docs.python.org/2/extending/extending.html>

<https://docs.python.org/2/c-api/veryhigh.html>

Both *graph\* g* and *hdl\_writer file\_writer* are already coded in the python shell. Simple commands used for testing:

> g.get\_num\_of\_nodes()

> file\_writer.write(g,libgraph.hdl\_writer.language.vhdl,"aes128\_fast.vhd")