UTILITIES

CALLING “C” FROM PYTHON

UNITTEST

PROFILING

Python Integration

- Python is in use at many places as an integration language, used to glue together ("steer") existing components.
- The strategy here is to create Python extension modules (written in C/C++) that make the functionality of large components written in C/C++ available to the Python programmer.
- The extension ("glue") modules are required because Python cannot call C/C++ functions directly

Python and “C/C++”

- Since Python is written in “C”, there is a facilities to link “C” code to Python.

- Several interfaces are available:
  - Ctype
  - Cffi
  - Subprocess
  - Python.h

CType

- Using the ‘ctype’ module several language options are available.
- The ‘ctype’ module is installed with Python.
- It provides C compatible data types, and allows calling functions in DLLs or shared libraries.

GCC

- Use GCC “GNU Compiler Collection” to create a compiled shared object:

```
gcc  -fPIC  -shared  -o  function.so  function.c
```

- Usage:

```
fun = ctypes.CDLL('function.so')
```

Windows DLL

- Windows has Dynamic Linked Libraries to link to “C” or “C++” code.
- Making a DLL a multi-step processes that can be built with a DLL wizard.
- Usage:

```
libc = ctypes.CDLL(‘Windows.dll’)
```

Python and Java Integration

- A new Python implementation written in 100% Pure Java, called Jython
- Jython offers seamless scripting for Java.
- It is a full implementation of the Python language and standard library, adding direct access to the universe of Java classes.
- Download: https://www.jython.org/download.html

Unit Testing

- Unit Testing is the first level of software testing where the smallest testable parts of a software are tested. This is used to validate that each unit of the software performs as designed.
- Python **unittest** module is used to test a unit of source code.
- To test your code you need to know what kind of data the function will return. After writing the code, you need to check whether the output is correct or not.

Unit Testing

- To have a complete set of manual tests, you make a list of the application features, input it can accept, and expected results. Every time you make a change, you need to go through every item on the list and re-check it.
- Unit Testing is the first level of software testing where the smallest testable parts of a software are tested. This is used to validate that each unit of the software performs as designed.
- A unit test is a smaller test, one that checks that a single component operates correctly. A unit test helps you to isolate what is broken in your application and fix it faster.

Profiling

- Software performance can be measured in a number of ways:
  - the problems it solves
  - its uptime
  - quantitative metrics
- Speed is probably the most important component of software performance.
- To measure the speed of Python code, a process called ‘profiling’ is used.

Speed Profiles

- Profiling measures the speed of your code and also provides additional information like:
  - The number of times each method was called
  - The average runtime of each method per call
  - The total time spent running each function

Profiling Information

- ncalls: the number of times each functions was called
- tottime: the total time spent running each function
- percall: the total time spent running the function divided by the number of calls
- cumtime: the total time spent running each function, as well as the functions that it itself called (in other words, its dependency functions)
- percall: cumtime divided by the number of calls

Python Profilers

- cProfile: a C extension suitable for profiling long-running programs. cProfile is recommend for most Python users.

- Profile: a pure Python module that adds significant technical overhead to profiled programs. Unless you're trying to extend your profiler beyond the functionality provided by cProfile, I would recommend against using the profile module.

- pstats: a Python module for formatting data and generating reports from the information generated by profile and cProfile.