Announcements

- Office Hours: Tuesdays 6–7 PM
- Send homework questions to mailing list
- Homework Submissions:
  - Programming Assignments: a hard-copy of the reports (only) must be submitted. An electronic copy of the report and the code must be submitted.
  - Paper Assignments: The entire printout must be submitted both as hard-copy and electronic copy.
Python Sample Program

- Implements a Stack
- There are three files:
  - StackDriver.py
  - IntStack.py
  - IntVector.py
StackDriver.py

- Imports IntVector and IntStack (part of this sample)
- Imports the ‘sys’ module: “provides a number of functions and variables that can be used to manipulate different parts of the Python runtime environment”
  - To get script arguments
  - To find built-in and imported modules
  - Among other things (can read more about this Online)
- This is the driver that manages input from the user
Python Sample Program

- IntStack.py
  - Imports IntVector
  - Implements the IntStack class
  - __init__ acts as a constructor that initializes the stack
  - ‘self’ is a variable that represents the instance of the object itself
  - Includes definitions of the push, pop, and size methods
Python Sample Program

- IntVector.py
  - Implements the IntVector class, which is a vector
  - Implements the get, set, resize, and size methods called by IntStack
Running the Sample Python Program on the CIMS Cluster

- Insert the following path to python in the python script: `#!/usr/local/bin/python` (didn’t have to add this to the Stack sample program, but in some cases you may need to add this)
- Run the python script with the following command: “python <script_name>”
- Run the Python program with the following command: “python StackDriver.py”
- Use the following commands (as an example) to manipulate the stack:
  - push 3
  - push 4
  - pop
  - quit
Installing Python on Windows

- Select the one compatible with your operating system and download the file (latest version of Python is 2.7)
- Install Python
- Setup the PATH environment variable:
  - In Control Panel, go to “System”
  - Click on the “Advanced” tab
  - Click on the “Environment Variable” button
  - Highlight PATH from the “System variables” section, and click on the “Edit” button
  - Append the path to the python executable to the “Variable value”, and continue to click “OK” (three times) until the System window goes away
Run the sample Python program:
- Open a DOS window by going to Start→Run, and entering cmd, and clicking “OK”
  - NOTE: you must set the PATH variable before you launch the DOS window otherwise it will not know the location of Python
- Navigate to the directory where you downloaded the Python sample program
- Run the program with the following command: “python StackDriver.py”
- Use the following commands (as an example) to manipulate the stack:
  - push 3
  - push 4
  - pop
  - quit
ADA Sample Program

- Implements a stack
  - There are three files
    - Driver.adb
    - Int_stacks.adb
    - Int_stacks.ads
Driver.adb

- Includes Int_Stacks
- The ‘with’ clause: declares that this program requires access to the Int_Stacks package. This makes the package visible to the program, but does not make its contents directly visible.
- The ‘use’ clause: makes the contents of the Int_Stacks package visible to this program
- This is the driver that manages input from the user
Int_stacks.adb and Int_stacks.ads
- Int_stacks.ads is the interface
- Int_stacks.adb is the implementation
  - Implements the push, pop, size methods, among others
  - Unchecked_Deallocation: generic procedure to release an object if it's no longer being referenced
Installing ADA on the CIMS Cluster

- ADA is installed on the following NYU servers (you must use one of these servers to use ADA):
  - courses1.cs.nyu.edu
  - courses2.cs.nyu.edu
Installing ADA on the CIMS Cluster

- Check which shell running in: `echo $SHELL`
  (on courses1.cs.nyu.edu, running bash shell)
- Set path environment variable for ADA:
  `PATH=/usr/local/pkg/gnat/bin:$PATH`
  - Verify that the path is properly set (the directory we added should be added to the beginning of the $PATH environment variable: `echo $PATH`
Set the following environment variables:
- **GCC_EXEC_PREFIX** = /usr/local/pkg/gnat/lib/gcc-lib/sparc-sun-solaris2.8/3.2.3/
- **ADA_INCLUDE_PATH** = /usr/local/pkg/gnat/lib/gcc-lib/sparc-sun-solaris2.8/3.2.3/adainclude:/usr/local/pkg/gnat/sources/gnat-gap-1.0.0-src/src
- **ADA_OBJECTS_PATH** = /usr/local/pkg/gnat/lib/gcc-lib/sparc-sun-solaris2.8/3.2.3/adalib
- **C_INCLUDE_PATH** = /usr/local/pkg/gnat/lib/gcc-lib/sparc-sun-solaris2.8/3.2.3/include

**** you can create a file (env-vals) in your home directory and run all of these commands at once by running the following command on the file env-vals:
```
source env-vals
```
Running the Sample ADA Program

- Compile the ADA program with the following command: `gnatmake driver.adb int_stacks.adb`
- Run the ADA program with the following command: `./driver`
- Use the following commands (as an example) to manipulate the stack:
  - push 3
  - push 4
  - pop
  - quit
Installing ADA on Windows

- Download ADA from the following Web site: http://www.mingw.org/