B Debugging in Python

“Don’t bug others, debug instead...”

Debugging

- Two types of errors occur while creating programs—Syntax errors and Logical errors.
- Syntax errors are grammatical errors and are reported by Python interpreter. It is easy to rectify these errors as interpreter tells us exactly which statement in the program is incorrect and why is it so.
- Logical errors are difficult to locate because we don't get any hint as to where things are wrong in our program and why we are not getting the desired results.
- Bug means an error. Debugging means process of removal of errors. Debugger is a special program the can help us detect Logical errors in our program.
- There are many debuggers available for debugging Python programs. No matter which debugger we use, the steps for debugging remain same. These steps are given below:

  (a) Start the debugger
  (b) Set breakpoints
  (c) Step through the source code one line at a time
  (d) Inspect the values of variables as they change
  (e) Make corrections to the source code
  (f) Rerun the program to make sure the fixes are correct

  Given below is a detailed explanation of these steps for IDLE debugger.

Start Debugger

- Start IDLE and type any program in it, or open an already typed source file.
- In the Shell window, click on the 'Debug' menu option at the top and then choose 'Debugger' from the pop-up menu. A new window shown in Figure B.1 and titled 'Debug Control' will appear on the screen.

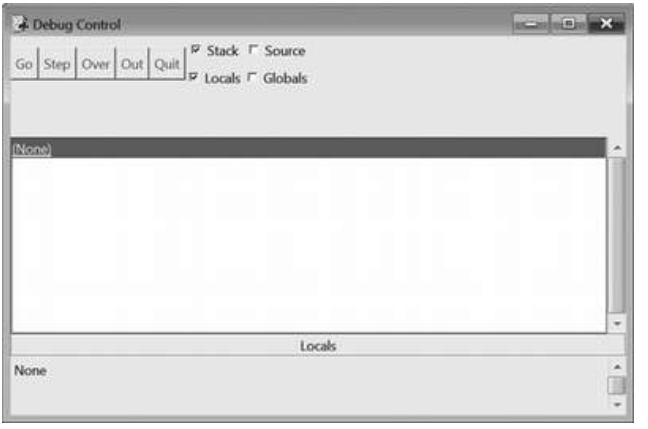

Figure B.1

- The Shell window will show:

```
>>>
[DEBUG ON]
>>>
```

Set Breakpoints

- A breakpoint is a marker in our code that tells the debugger that execution should proceed at normal speed up to the breakpoint, and stop there. Execution will not proceed beyond it unless we do so through manual intervention.
- Break points can be set in a program wherever we suspect something may go wrong. We can have many of them at different statements in one program.
- To set up a break point right click on a line of the source and choose 'Set breakpoint' from the menu.
- On setting a breakpoint the background color of the line turns yellow to show that a breakpoint has been set at that line.

Single Step through Program

- Execute the program using F5.
- The Debug Control window will now show in blue color the first line from where our program execution is to start. This means that line is ready to be executed.

- From this point we can click the 'Go' button in the Debug Control window to execute the program at normal speed until a breakpoint is encountered (or input is requested or the program finishes).
- Once control reaches the breakpoint, we can use the 'Step' button to step through our code, one line at a time. If the line being stepped through has a function call, execution will go to the first line of the function definition (we are "stepping into" the function). If we not wish to examine the statements in the function, we can choose the 'Over' button to step over the function.

Inspect Values

- As we single step through the program we can watch the type and value of local and global variables used in our program at the bottom of the Debug Control window.
- As different steps of our program get executed and the values of the variables change, the changed values get displayed in the Debug Control window.

Correct and Run Again

- By watching the values of the variables if we get a clue as to what is wrong with our program, we can stop the execution using the 'Quit' button. We can then rectify the program and debug it again using the same steps.
- While single stepping if we reach inside a function and we wish to finish execution of the function at normal speed and return from the function, we can do so using the 'Out' button.