CSI31 Introduction to Computer Programming I

Dr. Sharon Persinger
October24, 2018

Mathematical functions

- In math, a <u>function</u> is a rule that assigns to every element x of a set A, the domain of the function, one and only one element y of B, the codomain of the function.

- Examples:
  - $f(x) = 2x^2 + 3x + 1$
  - $g(x) = \sin(x)$

Functions in computer programming

main is a function

```
def main():
        #what follows is the definition of the main
        #function
main() #here we call or invoke main
```

Pick an example program.

Functions in computer programming

- •A function is a subprogram: a small program inside a program.
- •The function has a name.
- •The statements of the function can be executed by referring to the function name –calling the function or invoking the function.

Functions

- •Look at the program happy.py, chapter 6.

- •One function named happy.
- •Another function named sing.

- •sing has one <u>parameter</u>: person. A parameter is a variable that is given a value when the function is called.

- •Run happy, the program.
- •Import the functions in happy and run them.

Functions

- Compare a program written without functions to one written with functions.

  - futval_graph2.py
  - futval_graph4.py

Functions and parameters

- Functions can take parameters. drawBar takes three.
- Functions can take no parameters. createLabeledWindow takes no parameters
- •Why is the window a parameter for drawBar?
- •Variables used inside one function definition are local to that function. They are different from variables with the same name used in other functions.
- •The only way for a function to see a variable from another function is for the variable to be passed as a parameter.
- So the window must be passed to drawBar as a parameter.

Function definition syntax

- def <name> (<formal-parameters>):
  - <body>
- name is an identifier.
- Formal-parameters is a possibly empty list of variable names, alsoidentifiers.
- •body is a collection of statements.
- •The statements in body are indented.

Function call syntax

- <name> (<actual-parameters>)

- When Python gets to a function call:
  - The calling program is suspended at that point.
  - The formal parameters of the function get assigned the values given by the actual parameters used in the call.
  - The function body is executed.

- •Control returns to the calling function at the statement after the function call.

Return values

- Functions can do a task – drawBar draws a bar for the graph.

- A functions can compute a value.
- A functions can create an object.
  - For these two there must be a way to make what was created or computed available to the calling program. That is done by returning a value.

Look at the function createLabeledWindow. It creates a graphics window object and returns it to the calling program.

Why use functions?

- To simplify the structure of a program by breaking it into modules.
- To hide the details of a program by breaking it into modules.
- To allow multiple programmers to write pieces of a big software project.
- To be able to re-use code.

Write a function that computes present value.

- def presentvalue(endvalue, interestrate, years):
- You do the rest.