

Functional Decomposition: Extras

- Showing example code that you may need for Exercise 5, first option

James Tam

Parameter Passing And Return Values

Using predefined code e.g.,

```
roll = random.randrange(1,7)
```



Function returns information to be stored into variable 'roll'

Function definition 'def' must **store values** passed in

```
random - Notepad
File Edit Format View Help
def randrange(start, stop=None, step=_ONE):
    """Choose a random item from range(start, stop)
    This fixes the problem with randint() which i
```

Function body must **return a piece of information** back to call/running of function randrange.

```
version ,
DeprecationWarning, 2)
if stop is None:
    # We don't check for "step != 1" because
    # type checked and converted to an integer
    if step is not _ONE:
        raise TypeError('Missing a non-None st
    if istart > 0:
        return self._randbelow(istart)
```

James Tam

Example Of Parameter Passing, Returning Information

- Name of the full online example:

listParameterReturn.py

James Tam

Example Program: Passing Parameter

Calling function display,
pass in grades

```
def start():
    grades = [3.7,2.3,3.0,2.7]
    print("Pre-Pavol: ", end="\t")
    display(grades)
```

Function definition must **store values**
passed in.

```
def display(grades):
    for grade in grades:
        print(f"{grade:0.1f}",end=" ")
    print()
```

```
Pre-Pavol: 3.7 2.3 3.0 2.7
```

James Tam

Example Program: Function Returns A Value

Calling function
bestowBlessing...**pass in grades,**
store return result in variable
'grades'.

```
grades = bestowBlessingOfPavol(grades)
```

Function definition must **store values**
passed in.

```
def bestowBlessingOfPavol(grades):
```

Function body must **return a piece of**
information back to call/running of
function randrange.

```
    i = 0
    number = len(grades)
    while(i < number):
        grades[i] = grades[i] + BLESSING
        i = i + 1
    return(grades) #Return not needed (list)
```

Grades before/after the Pavol blessing

```
Pre-Pavol:      3.7 2.3 3.0 2.7
Post-Pavol:     3.8 2.4 3.1 2.8
..         ..
```

James Tam

Function Returning 'Many' Values

Function definition *apparently* (more
later) **returning 5 values.**

```
def countLetters(grades):
    aGrades = 0
    bGrades = 0
    cGrades = 0
    dGrades = 0
    fGrades = 0
    for grade in grades:
        if((grade>=MIN_A)):
            aGrades = aGrades + 1
        elif((grade>=MIN_B)):
            bGrades = bGrades + 1
        elif((grade>=MIN_C)):
            cGrades = cGrades + 1
        elif((grade>=MIN_D)):
            dGrades = dGrades + 1
        elif((grade>=MIN_F)):
            fGrades = fGrades + 1
        else:
            print("Invalid grade")
    return(aGrades, bGrades, cGrades, dGrades, fGrades)
```

Function call must use **5 variables to**
store all 5 values (there is a better
way you may see later – 'tuples')

```
first, second, third, fourth, fifth =  
    countLetters(grades)
```

James Tam