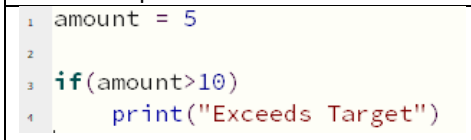
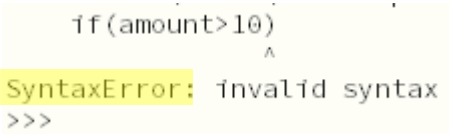
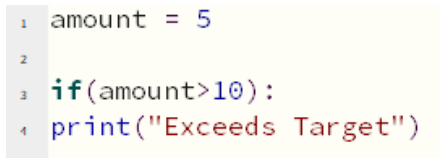
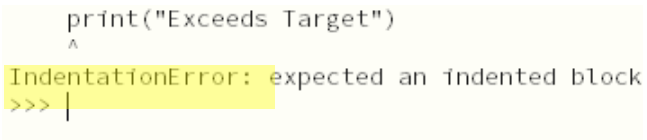
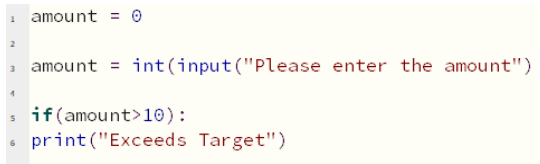
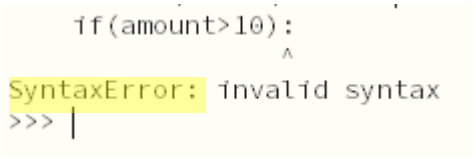
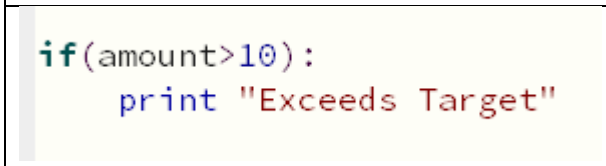
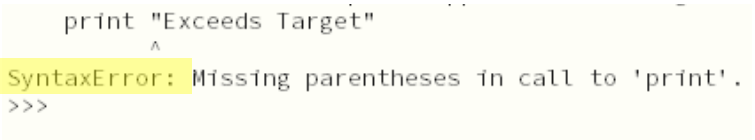
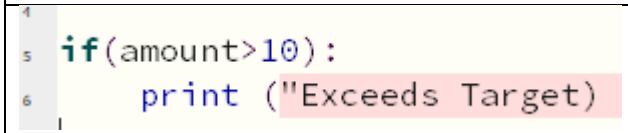
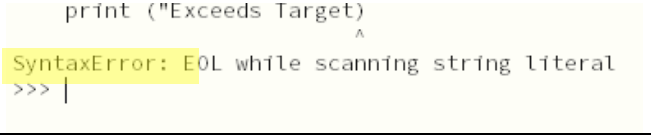
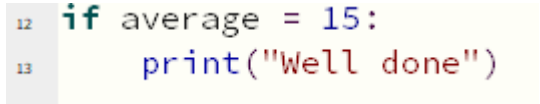
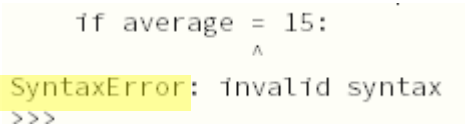
Python Common Error Codes – Debugging

Code Example	Error	Notes
		Missing colon :
		Constructs like if, loops and functions use indentation. Anything indented is included within the construct. Removing indent will end the construct.
		This can be confusing. The highlighted line is completely correct but the line above is missing a closing bracket. Always check the line above
		Parentheses means brackets. print output should always be inside brackets.
		Missing quotations to end string “Exceeds Target”)
		When comparing values python syntax uses double equals if average = = 15:

Running total with fixed loop – calculate average rainfall over 7 days using array `rainfall = [0] * 5` `total = 0` `average = 0` `for days in range(7):` `  rainfall[days] = int(input("Enter rainfall each day: "))` `  total = total + rainfall[days]` `average = total/7`	Execution Error - Index out of range Crashes during run `Enter rainfall each day: 3` `Enter rainfall each day: 4` `Enter rainfall each day: 2` `Enter rainfall each day: 6` `Enter rainfall each day: 7` `Enter rainfall each day: 2` `Traceback (most recent call last):` `  File "C:\ ...y", line 6, in <module>` `  rainfall[days] = int(input("Enter rainfall each day: "))` `IndexError: list assignment index out of range` `>>>`	It occurs when the Array Size is incorrect – not enough storage. Increase Array size needed. In this example, the array is only set to store 5 values and the fixed loop is looping 7 times. Program has nowhere to store the data so index out of range is range. Index just refers to the position within the array list. rainfall[0] You can use any variable to fill an array – doesn't have to be index all the time. This example uses [days]
Same example as above but Conditional loop. `rainfall = [0] * 5` `days = 0` `while days != 7:` `  rainfall[days] = int(input("Enter rainfall each day: "))` `  days = days + 1`	Execution Error - Index out of range Crashes during run `Enter rainfall each day: 2` `Enter rainfall each day: 3` `Enter rainfall each day: 4` `Enter rainfall each day: 5` `Enter rainfall each day: 6` `Enter rainfall each day: 4` `Traceback (most recent call last):` `  File "C:...or\.py"` `  rainfall[days] = int(input("Enter rainfall each day: "))` `IndexError: list assignment index out of range` `>>>`	days will continue to increment (+1) until it reaches 7 days – but the array has only been sized to store 5 list items.
`rainfall = [0] * 7` `total = 0` `average = 0` `for days in range(7):` `  rainfall[index] = int(input("Enter rainfal` `  total = total + rainfall[days]` `average = total/7`	Execution Error – Variable not defined Crashes during run `Enter rainfall each day2` `Traceback (most recent call last):` `  File "...s\Mu Editor\` `  rainfall[index] = int(input("Enter rainfall each day"))` `NameError: name 'index' is not defined` `>>> 2`	Not defined A Variable – in this case index - has not been defined. This fixed loop uses days to cycle through each time around the loop.

`` rainfall = [0] * 7 average = 0 for days in range(7): rainfall[days] = int(input("Enter r total = total + rainfall[days] average = total/7 ```	Execution Error – Variable not defined Crashes during run ``` Enter rainfall each day2 Traceback (most recent call last): File "(ms\Mu E total = total + rainfall[days] NameError: name 'total' is not defined >>> ```	total variable not defined total = 0 should be added to top.
`` rainfall = [0] * 7 average = 0 total = "" for days in range(7): rainfall[days] = int(input("Enter rainfall total = total + rainfall[days] average = total/7 ```	Execution Error – Variable type mismatch ``` Enter rainfall each day2 Traceback (most recent call last): File "C:\ ograms\Mu Edi total = total + rainfall[days] TypeError: can only concatenate str (not "int") to str >>> ```	total has been declared to store a string (text) but the program is asking the user to enter an integer and perform a calculation. Wrong data type
`` rainfall = [0] * 7 average = 0 total = 0 for days in range(7): rainfall[days] = int(input("Enter rain total = total + rainfall average = total/7 ```	Execution Error – Data type mismatch ``` Enter rainfall each day2 Traceback (most recent call last): File "C:\ s\Mu Editor total = total + rainfall TypeError: unsupported operand type(s) for +: 'int' and 'list' >>> ```	Data type mismatch. Array is declared but missing [variable] to cycle through array list. rainfall is not a single variable it is an array. rainfall[days]
Reading from File ``` #1. Read members data from walkers file. def readFileDate(members): counter = 0 with open("memberData.txt") as readFile: line = readFile.readline().rstrip("\n") while line: items = line.split(",") members[counter].forename = items[0] members[counter].surname = items[1] ```	File Not found ``` members = readFileDate(members) File "c: .uk\python prc readFileDate with open("memberData.txt") as readFile: FileNotFoundError: [Errno 2] No such file or directory: 'memberData.txt' >>> ```	Ensure that the text/csv file is in the same directory folder as your python code.

<table>
<tr>
<td>Functions/Procedures

```
# 2. Find the furthest distance walked.
def findLongestDistance(members):
furthest = members[0].distance
    for loop in range(1,len(members)):
        if members[loop].distance > furt
            furthest = members[loop].dis
    return furthest
```

</td>
<td>Syntax Error- Indentation

```
  File "                                    e - sout
    furthest = members[0].distance
    ^
IndentationError: expected an indented block
>>>
```

</td>
<td>When declaring functions/procedures code must be indented so the code is contained within the function/procedure.</td>
</tr>
<tr>
<td>Call statement – 3 Actual parameters

```
#2. Generate bib values and write to new file with entry IDs
generateBibValues(entryID,location,firstName)
```

4 formal parameters

```
def generateBibValues(entryID,location,firstName,surname):
    with open('bibValues.csv','w') as athletes:
        for loop in range(len(location)):
            bibValue = firstName[loop][0:1] + surname[loop]
            athletes.write(bibValue + "\n")
```

</td>
<td>Parameter mismatch

```
Traceback (most recent call last):
  File "c:\us                                          hon programs\p
    generateBibValues(entryID,location,firstName)
TypeError: generateBibValues() missing 1 required positional argument: 'surname'
>>>
```

</td>
<td>Arguments is another name for parameters.

There are 3 actual parameters in the call statement but 4 formal parameters.

Formal and Actual parameters can be different names but the order in which they are passed is vitally important.</td>
</tr>
<tr>
<td>

```
#3. Function to find the highest numbe
maxJumps = findMax(jumps)
```

```
# 3. Find max algorithm to find the highest nur
def findMax(jumps):
    maxJumps = jumps[0]
    for loop in range(1,len(jumps)):
        if jumps[loop] > maxJumps:
            maxJumps = jumps[loop]
```

</td>
<td>No error but unexpected results.

No value is being returned from the findMax function</td>
<td>return maxJumps

should be added to the end of the function to ensure the correct data is passed out to the main program.</td>
</tr>
</table>