IHF: CODE

PYTHON – SESSION 4

REVIEW

MODULES

```
import random
from math import floor
```

RANDOM MODULE

```
import random

# Random float from 0.0 to 1.0
print random.random()

# Gets a random number between 1 and 10
number = random.randint(1, 10)
```

MATH MODULE

```
from math import floor, ceil

number = floor(3.2) # 3
print(floor(9.99)) # 9

number = ceil(3.2) # 4
print(ceil(9.99)) # 10
```

WHILE LOOPS

```
guess = None
while guess != 4:
    # Continues to ask for a number until you enter 4
    guess = int(input("What's your number? "))
```

INFINITE LOOPS

```
while True:
    # This loops forever
    print("Hello")
```

BREAK STATEMENTS

```
while True:
    print("Hello")
    break
```

COLLECTIONS

- List
- Tuple
- Set
- Dictionary

COLLECTIONS – TUPLE

```
colours = ("Red", "Blue", "Green")
print(colours[0]) # Red
print(colours[1]) # Blue
print(colours[2]) # Green
```

COLLECTIONS – SET

```
fruit = {"Apple", "Banana", "Cherry"}
for item in fruit:
    print(item)
```

COLLECTIONS – DICTIONARY

```
shirt = {
    "size": "Large",
    "colour": "Red"
}

print(shirt["size"]) # Large

shirt["material"] = "Cotton" # Add new key/value pair
shirt["colour"] = "Green" # Change existing value

del(shirt["size"]) # Delete key/value pair

if "material" in shirt:
    print("The material is: " + shirt["material"])

for key in shirt:
    print(str(key) + " = " + str(shirt[key]))
```

NESTED COLLECTIONS

```
phone_grid = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9],
    ["*", 0, "#"]
]

for row in phone_grid:
    for column in row:
        print(column)
```

LIST OF DICTIONARIES

```
contacts = [
    {"fname": "Alice", "lname": "Smith"},
    {"fname": "Bob", "lname": "Jones", "phone": "555-1234"},
    {"fname": "Charlie", "lname": "McCloud"}
]

for person in contacts:
    if "phone" in person:
        print(person["fname"])
```

QUESTIONS?

FUNCTIONS

FUNCTIONS – CREATE

```
def hello_world():
    print("Hello World!")
```

FUNCTIONS – CREATE

```
def <function_name>():
    <your code here>
```

FUNCTIONS – CALL

```
def hello_world():
    print("Hello World!")

hello_world()
```

FUNCTIONS – PARAMETERS

```
def hello(name):
    print("Hello, " + name + "!")

hello("Alice")
hello("Bob")
hello("Charlie")
```

FUNCTIONS – PARAMETERS

```
def hello(name, age):
    print("Hello my name is " + name)
    print("I'm " + str(age) + " years old")

    age_in_10_years = age + 10
    print("In 10 years time I will be " + str(age_in_10_years))

hello("Alice", 22)
hello("Bob", 34)
hello("Charlie", 17)
```

FUNCTIONS – PARAMETERS

```
def area(x, y, z):
    print("The area is " + str(x * y * z))

area(12, 3, 4)
area(6, 14, 10)
```

FUNCTIONS – PARAMETERS

```
def <function_name>(<param_1>, <param_2>, ...):
    <your code here>
```

CODING TIME

SECTION A

FUNCTIONS – RETURNING

```
def area(x, y, z):
    return x * y * z

cube1 = area(12, 3, 4)
cube2 = area(6, 14, 10)
```

FUNCTIONS – RETURNING

```
def <function_name>(<param_1>, <param_2>, ...):
    <your code here>

    return <value>
```

FUNCTIONS – SINGLE JOB

```
def hello(name, age):
    print("Hello my name is " + name)
    print("I'm " + str(age) + " years old")
    print("In 10 years time I will be " + str(age_in_x_years(age, 10)))

def age_in_x_years(age, years):
    return age + years

hello("Alice", 22)
hello("Bob", 34)
hello("Charlie", 17)
```

FUNCTIONS – RECURSION

```
def calc_factorial(x):
    if x == 1:
        return 1
    else:
        return (x * calc_factorial(x - 1))

num = 4
print("The factorial of " + str(num) + " is " + str(calc_factorial(num)))
```

FUNCTIONS – RECURSION

```
def calc_factorial(x):
    if x == 1:
        return 1
    else:
        return (x * calc_factorial(x - 1))

# calc_factorial(4)                 # 1st call with 4
# 4 * calc_factorial(3)             # 2nd call with 3
# 4 * 3 * calc_factorial(2)         # 3rd call with 2
# 4 * 3 * 2 * calc_factorial(1)     # 4th call with 1
# 4 * 3 * 2 * 1                     # return from 4th call as number=1
# 4 * 3 * 2                         # return from 3rd call
# 4 * 6                             # return from 2nd call
# 24                                # return from 1st call
```

CODING TIME

SECTION B

EXERCISES

Finish off any exercises you did not complete in the session

FURTHER HELP

DL-UKIHFCODE@KPMG.CO.UK