Ministry of Higher Education

Kabul University

Faculty of Information Technology and Telecommunications

Department of Information Science and Engineering (ISE)

Fourth home work of Introduction to Python Programming:

Multiple Choice Question:

1) A local variable in Python is a variable that is,

   A) Defined inside every function B) Local to the given program C) Accessible from within the function D) All of these

2) Which of the following statements are the advantages of using functions?

   A) Reduce duplication of code B) Clarity of code C) Reuse of code D) All of these

3) The keyword that is used to define the block of statements in function?

   A) function B) func C) def D) pi

4) The characteristics of docstrings are

   A) suitable way of using documentation B) Function should have a docstring C) Can be accessed by __doc()__ D) All of these

5) The two types of functions used in Python are

   A) Built-in and user-defined B) Custom function and user function C) User function and system call D) System function

6) _____ refers to built-in mathematical function.

   A) sqrt B) rhombus C) add D) sub

7) The variable defined outside the function is referred as

   A) static B) global C) automatic D) register

8) Functions without a return statement do return a value and it is

   A) int B) null C) None D) error

9) The data type of the elements in sys.argv?

   A) set B) list C) tuple D) string

Present by: Habibullah Mdadi

10) The length of sys.argv is?

A) Total number of arguments excluding the filename B) Total number of arguments including the filename C) Only filename D) Total number of arguments including Python Command

11) The syntax of keyword arguments specified in the function header?

A) * followed by an identifier B) _ followed by an identifier C) ** followed by an identifier D) _ _ followed by an identifier

12) The number of arguments that can be passed to a function is

A) 0 B) 1 C) 0 or more D) 1 or more

13) The library that is used to create, manipulate, format and convert dates, times and timestamps in Python is

A) Arrow B) Pandas C) Scipy D) NumPy

14) The command line arguments is stored in

A) os.argv B) sys.argv C) argv D) None

15) The command that is used to install a third-party module in Python is

A) pip B) pipe C) install_module D) pypy

16) . Judge the output of the following code.

```
import math
math.sqrt(36)
```

A) Error B) −6 C) 6 D) 6.0

17) . The function divmod(10,20) is evaluated as

A) (10%20,10//20) B) (10//20,10%20) C) (10//20,10*20) D) (10/20,10%20)

18) Predict the output of the following code?

```
def tweet():
                print("Python Programming!")
tweet()
```

A) Python Programming! B) Indentation Error C) Syntax Error D) Name Error

19) The output of the following code is

```
def displaymessage(message, times = 1):
        print(message * times)
displaymessage("Data")
displaymessage("Science", 5)
```

A) Data Science Science Science Science Science B) Data Science 5

C) DataDataDataDataDataScience D) DataDataDataDataDataData

20) . Guess the output of the following code

```
def quad(x):
  return x * x * x * x
```

```
x = quad(3)
print(x)
```

A) 27 B) 9 C) 3 D) 81

21) The output of the following code is

```
def add(*args):
    x = 0
    for i in args:
        x += I
    return x
print(add(1, 2, 3))
print(add(1, 2, 3, 4, 5))
```

A) 16 15 B) 6 15 C) 1 2 3 D) 1 2 3 45

22) Gauge the output of the following code.

```
def foo():
    return total + 1
total = 0
print(foo())
```

A) 1 B) 0 C) 11 D) 00

23) The default arguments specified in the function header is an

A) Identifier followed by an = and the default value B) Identifier followed by the default value within back-ticks C) Identifier followed by the default value within [] D) Identifier followed by an #.

Review Questions:

1) Define function. What are the advantages of using a function?

   Answer: A function is a named block of code that performs a specific task, optionally accepts inputs (parameters), and can return a value. Function encapsulate logic and allow reuse.

   Advantages:

   - Reusability: Write once, call many times.
   - Modularity: Break program into logical units.
   - Readability: Names clarify intent.
   - Maintainability: Easier to test and update isolated code.
   - Abstraction: Hide implementation details.

2) Differentiate between user-defined function and built-in functions.

   Answer:

   - Built-in function: provided by the language (e.g., len(),sum()). They are ready to use and optimized.

- ❖ User-defined function: written by the programmer to implement domain-specific logic. They enable custom behavior that built-ins don't provide.

3) Explain with syntax how to create a user-defined functions and how to call the user-defined function from the main function.

Answer:

```
def great(name):
    """return greeting string for name."""
    Return f" Hello, {name}!"
def main():
     s = greet("Alice")
user-defined function
print(s)
if _name_ == "_main":
     main()
```

Explanation: define with def name (params): …. Call by using the function name and passing arguments.

4) Explain the built-in functions with examples in Python.

Answer:

```
Print(len([1,2,3]))
Print(sum([1,2,3]))
Print(max(3,7,1))
Print(min([4,2,9]))
Print(sirted([3,1,2]))
Print(list(map(str,[1,2,3])))
Print(list(filter(lambda x: x%2==0,[1,2,3,4])))
Print(types(5))
Print(isinstance(5, int))
```

(There are many more built-ins: open, range, enumerate, zip, etc.)

5) Differentiate between local and global variables with suitable examples.

Answer:

Global variable: declared at modules level; accessible anywhere in the modules (unless shadowed).

Local variable: defined inside a function; scope limited to that function.

Ex:

```
X = 10
def() f:
```

```
    y = 5
print("inside f:",x,y)
f()
print("outside:" , x)
```

If you need to modify a global variable inside a function, use global (but prefer returning values instead of using globl).

6) Explain the advantages of *args and **kwargs with examples.

Answer:

- *args allows a function to accept any number of positional arguments(packed as a tuple).
- **kwargs allows any number of keyword argument (packed as a dict). They make function flexible and forwardable.

Ex:

```
def summer(*args):
    return sum(args)
def printer(**kwargs):
    for k, v in kwargs.items():
        print(k, "=",v)
print(summer(1,2,3))
printer(name = "Alice" , age = 30)
```

They are useful for wrapper functions, APIs, and default-forwarding.

7) Demonstrate how functions return multiple values with an example.

Answer:

Python function can return a tuple; caller can unpack.

Ex:

```
def stats(numbers):
    total = sum(numbers)
    count = len(numbers)
    mean = total / count if count else None return total, count, mean
    t, c, m = stats([1,2,3,4])
    print(t, c, m)
```

8) Explain the utility of docstrings?

Answer:

A docstring is a string literal placed right after a function/class/module definition that documents its purpose, parameters, and return values. Accessible at runtime via. __doc__and used by IDEs and help other developers and yourself.

Ex:

```
def add(a, b):
    return the sum of a and b”””””””(number).
    Return a + b
Print(add._doc__)
```

9) Write a program using functions to perform the arithmetic operations.

Answer:

```
def add(a,b): return a+b
def add(a,b): return a-b
def add(a,b): return a*b
return a/b if b! = 0 else float(‘inf’)
def mod(a,b): return a % b
def mod(a,b): return a ** b
def mod(a,b): return a // b
def main():
    a, b = 12, 5
print(“add:”, add(a,b))
print(“sub:”, sub(a,b))
print(“mul:”, mul(a,b))
print(“div:”, div(a,b))
print(“mod:”, mod(a,b))
print(“power:”, power(a,b))
print(“floor div:”, floor div(a,b))
if __name__ == “__main__ “:
    main()
```

10) Write a program to find the largest of three numbers using functions.

Answer:

```
def largest_of_three(a,b,c):
    return max(a,b,c)
print(largest_of_three(3,7,5))
```

11) Write a Python program using functions to find the value of nPr and nCr.

Answer:

```
def factorial(n):
    if n < 0:
        raise valueError(“n must be non-negative”)
    result = 1
    for i in range (2, n+1):
        result *= i
def npr(n, r):
if r > n: return 0
return factorial(n) // factorial(n-r)
def nCr(n,r):
    if r > n: return 0
r = min(r, n-r)
num = 1
den = 1
for i in range(1, r+1):
    num *= n – r + i
    den *= i
return num // den
print(nPr(5, 3))
print(nCr(5,3))
```

12) Write a Python function named area that finds the area of a pentagon.

Answer:

Assume regular pentagon with sight length.

A formula for area:

$$\text{Area} = \frac{1}{4}\left(\sqrt{5(5+2\sqrt{5})}\right) a^2$$

Code:

```
Import math
def area_pentagon(side):
     return area of a regular pentagon”””””””. ‘With sight length side’
     factor = 0.25 * math.sqrt(5*(5 + 2 *  math.sqrt(5)))
     return factor * side * side
print(area_pemtagon(3.0))
```

- ❖ (If non_regular pentagon. Area needs coordinates or apothem.)

13) Write a program using functions to display Pascal's triangle.

Answer:

```
def pascal_triangle(n_rows):
    returns list of rows of pascal 's""" """. Triangle up to n_rows (n_rows >= 1)
    [] = rows
    For i in range(n_rows):
        if i == 0
    row = [1]
else:
     prev = rows[-1]
     row = [1] + [prev[j] + prev[j+1] for j in range(len(prev) -1)] + [1]
     rows.append (rows)
return rows
for row in pascal_triangle(6):
print(row)
```

14) Write a program using functions to print harmonic progression series and its sum till N terms.

Answer:

```
def harmonic_progression(a, d, n):
    Return list of first n terms of HP""" """. and their sum; HP: 1/(a + k*d)
    [] = terms
    S = 0.0
    For k in range(n):
        Denom = a + k*d
        If denom == 0:
            Raise
ZeroDivisionError("Denominator becomes zero in HP")
Terms = 1.0 / denom
Terms.append(term)
S += term
Return terms, s
Terms, total = harmonic_pentagon(1, 1, 5)
Print (terms
Print ("sum =", total)
```

15) Write a program using functions to do the following tasks:

A) Convert milliseconds to hours, minutes and seconds.

B) Compute a sales commission, given the sales amount and the commission rate.

C) Convert Celsius to Fahrenheit.

D) Compute the monthly payment, given the loan amount, number of years and the annual interest rate.

Answer:

A:

```
def ms_to_hms(ms):
    seconds_total = ms // 1000
    hourse = seconds_total // 3600
    minutes = (second_total % 3600) // 60
    seconds = seconds_total % 60
    return hours, minute, seconds
print(ms_to_hms(3_660_500))
```

B: Compute sales commission given amount and rate.

EX:

```
compute sales commission given amount and rate.
def commission(sales_amount, rate_precent):
    return commission values. “”” “””. Rate_precent is like 5 for 5%
    return sales_amount * (rate_precent / 100.0)
print(commission (100.0))
```

C: Convert Celsius to Fahrenheit.

EX:

```
def c_to_f(c):
   return (c * 9 / 5) + 32
print(c_to_f(0))
```

D: Compute monthly payment for a loan.

EX:

```
def monthly_payment(loan_amount, years, annual_rate_percent):
    return monthly payment. “”” “””. Annual_rate_percent e.g. 6.5 for 6.5%
    n = years * 12
    r = (annual_rate_percant / 100.0 / 12.0)
    if r == 0:
        return loan_ amount / n
return loan) amount * (r / (1 – (1 + r ** -n)))
```

```
print(monthly_payment(100000, 15, 6.5))
```

By Respect: Habibullah Madadi