ASSIGNMENT-1

Page no.: ____/
Date: ___/___/____

1] Define all data types in python also provide 5 functions for each Data Type with example.

Ans:- Python has various data types. Here is a summary with 5 functions for each type:

a. Numeric Types (int, float, complex)

- int: integer values.
- float: Decimal numbers.
- complex: Numbers with a real and imaginary part.

- Examples of Functions:-

```
#int
num = 42
print(num.bit_length()) #Number of bits required to
                        represent 42 in binary.
print(num.__add__(8)) # Add 8 to 42
print(num.__mul__(2)) # multiply by 2
print(num.to_bytes(4, 'big')) # Convert to 4 bytes
print(int("1010", 2))  # Convert binary string to
                         int

#float
flt = 42.42
print(round(flt))      #Round to nearest integer
print(flt.is_integer()) # check if it is an integer.
print(flt.__add__(2.58)) # Add another float
print(abs(-flt))       # absolute value
print(format(flt, ".2f")) # Format with 2 decimal
                            points.
```

Page no.: ____/
Date: ____/____/____

```
# complex
comp = 3 + 4j
print(comp.real)          # Real part
print(comp.imag)          # imaginary part.
print(abs(comp))          # Magnitude
print(comp.conjugate())   # conjugate
print(comp.__mul__(2))    # Multiply by a scalar.
```

b. String (str)

- Examples of Functions:

```
txt = "Hello Python"
print(txt.upper())        # Convert to uppercase
print(txt.lower())        # Convert to lowercase
print(txt.split())        # Split into a list.
print(len(txt))           # Length of the string
```

c. Sequence Types (list, tuple, range)

- List Example Functions:

```
lst = [1, 2, 3, 4]
lst.append(5)             # Append element
lst.pop()                 # Remove last element
lst.reverse()             # Reverse the list.
print(lst.count(2))       # Count occurrences of 2
print(lst.index(3))       # index of 3
```

Page no.:
Date:

- Tuple Example Functions:

```
tpl = (1, 2, 3)
print (len(tpl))          # Length of tuple
print (tpl.index(2))      # Index of 2
print (tpl.count(1))      # count occurences of 1
tpl2 = tpl + (4,)         # count concatenation
print(tpl2)
```

d. Mapping Type (dict)

- Examples of Functions:

```
d = {'a': 1, 'b': 2}
print (d.keys())          # Get all keys
print (d.values())        # Get all values
print (d.get('a'))        # Get value by key
d.update ({'c': 3})       # Update with another dict.
print (d.pop ('b'))       # Remove and return value for
                            'b'
```

e. Set Types (set, frozenset)

```
s = {1, 2, 3}
s.add(4)                  # Add element
s.remove(2)               # Remove element
print (s.union {5, 6}))   # Union with another set
print (s.intersection {3, 4, 5}))   # Intersection
print (s.difference ({1, 4}))       # Diffrence.
```

Page no.: ____/
Date: __/__/____

Q2] What are jump statement in python also give their examples.

Ans Jump statements alter the flow of control. Examples:

1. break: Exits the loop entirely.
2. continue: Skips the current iteration.
3. pass: Placeholder with no action.
4. return: Exits a function.
5. raise: Raises an exception.

- Examples:-

```
#break
for i in range(5):
    if i == 3:
        break
    print(i)

#continue
for i in range(5):
    if i == 3:
        continue
    print(i)

#pass
for i in range(5):
    if i == 3:
        pass
```

Page no.:
Date: / /

```
print(i)

# return
def func():
    return 42
print(func())

# raise
try:
    raise ValueError("An error occurred!")
except ValueError as e:
    print(e)
```

Q3] Define memory allocation in python. Also write python is interpreted language.

Ans: Memory Allocation in Python

- Heap Memory: Objects and data are stored in the heap. Python's garbage collector manages memory.
- Stack Memory: Function calls and local variables are stored in the stack.

Python as an Interpreted Language:

- Python code is executed line by line, converting it to machine code dynamically. This makes it slower than complied languages but allows flexibility and easier debugging.

Page no.: ____/
Date: __/__/____

Assignment - 2

Q 1. What is function in python? Give syntax. What is different type of argument in Python. Give examples of combination and keyword argument.

Ans] Functions in Python

→ A function is a block of reusable code.

→ Syntax:

```
def function_name(parameters):
    #Function body
    return value.
```

Types of Arguments:

1. Default
2. Positional
3. Keyword
4. Arbitrary (*args, **kwargs)

• Example:-

```
def func(a, b=10, *args, **kwargs):
    print(a, b, args, kwargs)
func(1, 2, 3, 4, x=5, y=6) #Positional, default,
                            arbitrary, keyword.
```

Page no.:____/
Date:__/__/____/

Q2] What is lamda expression, define special function of lamda function.

Ans:- A lamda is an anonymos function.
Special Feature: Single-expression functions.

- Example:-

```
square = lamda x: x ** 2
print (square(5))

#Using lamda in 'sorted'
data = [('a', 2), ('b', 1)]
sorted_data = sorted(data, key = lamda x: x[1])
print (sorted_data)
```

Q3] What is list comprehension, also provide 4 examples of list comprehension.

Ans:- Concise way to create lists.

- Example:-

```
#Square of numbers
squares = [x ** 2 for x in range(5)]
print (squares)

#Even numbers
evens = [x for x in range(10) if x % 2 == 0]
print (evens pairs)
```

Page no.:
Date: / /

Transforms strings

upper = [char.upper() for char in 'hello']
print(upper)

Q4] Why we can't store dictionary in set, also provide example.

Ans:- Sets store immutable data, and dictionaries are mutable (keys can change).

Example:-

```
# This will raise a TypeError.
try:
    s = {1, 2, {3: 'value'}}
except TypeError as e:
    print(e)
```