**Discussion 1**

What is the output of the following program?

sample\_string = "Hello, World! Welcome to Python programming."

upper\_string = sample\_string.upper()

print(f"Uppercase: {upper\_string}")

lower\_string = sample\_string.lower()

print(f"Lowercase: {lower\_string}")

capitalized\_string = sample\_string.capitalize()

print(f"Capitalized: {capitalized\_string}")

position = sample\_string.find("World")

print(f"Position of 'World': {position}")

replaced\_string = sample\_string.replace("World", "Universe")

print(f"Replaced string: {replaced\_string}")

words\_list = sample\_string.split()

print(f"List of words: {words\_list}")

joined\_string = " ".join(words\_list)

print(f"Joined string: {joined\_string}")

The output of that program is:

Uppercase: HELLO, WORLD! WELCOME TO PYTHON PROGRAMMING.

Lowercase: hello, world! welcome to python programming.

Capitalized: Hello, world! welcome to python programming.

Position of ‘World’: 7 //This is the index of W

Replaced string: Hello, Universe! Welcome to Python programming.

List of words: ['Hello,', 'World!', 'Welcome', 'to', 'Python', 'programming.'] //All the signs are incorporated into the words

Joined string: Hello, World! Welcome to Python programming.

**Discussion 2**

What is the output of the following program?

temperatures = [72, 68, 75, 70, 69, 74, 73]

print(f"Recorded temperatures: {temperatures}")

temperatures.sort()

print(temperatures)

new\_temperature = 71

temperatures.append(new\_temperature)

print(f"After appending new temperature {new\_temperature}: {temperatures}")

max\_temperature = max(temperatures)

print(f"The maximum temperature recorded is: {max\_temperature}")

min\_temperature = min(temperatures)

print(f"The minimum temperature recorded is: {min\_temperature}")

pre\_temperatures = [51, 53, 56]

pre\_temperatures.append(temperatures)

print(f"after apend pre\_temperatures:{pre\_temperatures}")

aft\_temperatures = [95, 97, 94]

for temp in aft\_temperatures:

temperatures.append(temp)

print(f"After appending temperatures for another week: {temperatures}")

average\_temperature = sum(temperatures)/len(temperatures)

print("The average temperature is:", average\_temperature)

The output of that program is:

Recorded temperatures: [72, 68, 75, 70, 69, 74, 73]

[68, 69, 70, 72, 73, 74, 75]

After appending new temperature: [68, 69, 70, 72, 73, 74, 75, 71]

The maximum temperature recorded is 75

The minimum temperature recorded is 68

After append pre\_temperatures: [68, 69, 70, 72, 73, 74, 75, 71, 51, 53, 56]

After appending temperatures for another week: [68, 69, 70, 72, 73, 74, 75, 71, 51, 53, 56, 95, 97, 94]

The average temperature is 78.0

**Discussion 3**

What is the output of the following program?

# Define a list of lists representing the weekly schedule for employees

weekly\_schedule = [

["Meeting", "Emails", "Project Work", "Lunch", "Code Review", "Call with Client", "Report Writing"], # Employee 1

["Code", "Emails", "Design", "Lunch", "Testing", "Meeting", "Documentation"], # Employee 2

["Research", "Emails", "Development", "Lunch", "Planning", "Team Meeting", "Presentation"], # Employee 3

]

l1 = weekly\_schedule[0][:3]

print(l1)

l2 = weekly\_schedule[1][-3:]

print(l2)

l3 = weekly\_schedule[2][4]

print(l3)

The output of that program is:

[“Meeting”, “Emails”, “Project Work”]

[“Testing”, “Meeting”, “Documentation”]

Planning

**Discussion 4**

Write a Python program, in the fewest number of lines possible, which creates a list of all the square numbers: x2 (where 1<=x<=100) that are divisible by 3.

squares = []

for x in range(1, 101):

    if x % 3 == 0:

        squares.append(x\*\*2)

print(squares)

**Discussion 5**

**What is the output of the following program?**

# Define a dictionary representing a collection of books in a library

library = {

"978-0143127741": {

"title": "To Kill a Mockingbird",

"author": "Harper Lee",

"year": 1960,

"genre": "Fiction"

},

"978-0439023481": {

"title": "The Hunger Games",

"author": "Suzanne Collins",

"year": 2008,

"genre": "Dystopian"

},

"978-0307277671": {

"title": "The Road",

"author": "Cormac McCarthy",

"year": 2006,

"genre": "Post-apocalyptic"

}

}

# Print the entire library dictionary

print("Library collection:")

for isbn, book in library.items():

print(f"ISBN: {isbn}, Book: {book}")

# Access information about a specific book by its ISBN

isbn\_to\_lookup = "978-0143127741"

book\_info = library.get(isbn\_to\_lookup, "Book not found")

print(f"\nDetails of the book with ISBN {isbn\_to\_lookup}: {book\_info}")

# Add a new book to the library

new\_book\_isbn = "978-0553573404"

new\_book = {

"title": "A Game of Thrones",

"author": "George R. R. Martin",

"year": 1996,

"genre": "Fantasy"

}

library[new\_book\_isbn] = new\_book

print(f"\nAdded new book with ISBN {new\_book\_isbn}: {library[new\_book\_isbn]}")

# Remove a book from the library

isbn\_to\_remove = "978-0307277671"

removed\_book = library.pop(isbn\_to\_remove, "Book not found")

print(f"\nRemoved book with ISBN {isbn\_to\_remove}: {removed\_book}")

# Update the details of an existing book

isbn\_to\_update = "978-0439023481"

library[isbn\_to\_update]["year"] = 2009 # Updating the year of publication

print(f"\nUpdated book details with ISBN {isbn\_to\_update}: {library[isbn\_to\_update]}")

# Search for books by a specific author

author\_to\_search = "Harper Lee"

books\_by\_author = [book for book in library.values() if book["author"] == author\_to\_search]

print(f"\nBooks by {author\_to\_search}: {books\_by\_author}")

# Filter books by genre

genre\_to\_filter = "Fiction"

books\_in\_genre = {isbn: book for isbn, book in library.items() if book["genre"] == genre\_to\_filter}

print(f"\nBooks in the genre '{genre\_to\_filter}': {books\_in\_genre}")

The output of this program is:

ISBN: 978-0143127741, Book: {'title': 'To Kill a Mockingbird', 'author': 'Harper Lee', 'year': 1960, 'genre': 'Fiction'}

ISBN: 978-0439023481, Book: {'title': 'The Hunger Games', 'author': 'Suzanne Collins', 'year': 2008, 'genre': 'Dystopian'}

ISBN: 978-0307277671, Book: {'title': 'The Road', 'author': 'Cormac McCarthy', 'year': 2006, 'genre': 'Post-apocalyptic'}

Details of the book with ISBN 978-0143127741: {'title': 'To Kill a Mockingbird', 'author': 'Harper Lee', 'year': 1960, 'genre': 'Fiction'}

Added new book with ISBN 978-0553573404: "title": "A Game of Thrones", "author": "George R. R. Martin", "year": 1996, "genre": "Fantasy"}

Removed book with ISBN 978-0307277671: {'title': 'The Road', 'author': 'Cormac McCarthy', 'year': 2006, 'genre': 'Post-apocalyptic'}

Updated book details with ISBN 978-0439023481: {'title': 'The Hunger Games', 'author': 'Suzanne Collins', 'year': 2009, 'genre': 'Dystopian'}

Books by Harper Lee: [{'title': 'To Kill a Mockingbird', 'author': 'Harper Lee', 'year': 1960, 'genre': 'Fiction'}]

Books in the genre ‘Fiction:’ {'978-0143127741': {'title': 'To Kill a Mockingbird', 'author': 'Harper Lee', 'year': 1960, 'genre': 'Fiction'}}