

#Task_1:

```
def manage_courses():
    courses = {
        'CSE101': {'Course name': 'Introduction to Programming', 'Credits': 3, 'Instructor': 'Dr. Alice'},
        'CSE102': {'Course name': 'Data Structures', 'Credits': 4, 'Instructor': 'Dr. Bob'},
        'CSE103': {'Course name': 'Database Systems', 'Credits': 3, 'Instructor': 'Dr. Carol'}
    }
```

```
courses['CSE102']['Instructor'] = 'Dr. Bob Jr.'
```

```
courses['CSE104'] = {'Course name': 'Algorithms', 'Credits': 4, 'Instructor': 'Dr. Dave'}
```

```
del courses['CSE101']
```

```
for course_code, details in courses.items():
    print(f"{course_code}: {details}")
```

```
manage_courses()
```

```

CSE102: {'Course name': 'Data Structures', 'Credits': 4, 'Instructor': 'Dr. Bob Jr.'}
CSE103: {'Course name': 'Database Systems', 'Credits': 3, 'Instructor': 'Dr. Carol'}
CSE104: {'Course name': 'Algorithms', 'Credits': 4, 'Instructor': 'Dr. Dave'}
```

#Task_2:

```
def string_operations():
```

```
    sentence = "Learning Python is fun and rewarding."
```

```
    substring = sentence[-31:-12]
    print(f"Extracted substring: {substring}")
```

```
    modified_sentence = sentence.replace("rewarding", "exciting")
    print(f"Modified sentence: {modified_sentence}")
```

```
    position = modified_sentence.index("exciting") + len("exciting")
    final_sentence = modified_sentence[:position] + " Keep practicing!" + modified_sentence[position:]
    print(f"Final sentence after insertion: {final_sentence}")
```

```
    capitalized_sentence = final_sentence.title()
    print(f"Capitalized sentence: {capitalized_sentence}")
```

```
string_operations()
```

```

Extracted substring: ng Python is fun an
Modified sentence: Learning Python is fun and exciting.
Final sentence after insertion: Learning Python is fun and exciting Keep practicing!.
Capitalized sentence: Learning Python Is Fun And Exciting Keep Practicing!.
```

#Task_3:

```
def manage_customers():
```

```
    customers = ["Alice", "Bob", "Charlie", "David", "Eve"]
```

```
    third_customer = customers[2]
    print(f"Third customer: {third_customer}")
```

```
    customers[1] = "Ben"
```

```
    customers.append("Frank")
```

```
    customers.remove("David")
```

```
    customers.sort()
    print(f"Final sorted customer list: {customers}")
```

```
manage_customers()
```

```
→ Third customer: Charlie
Final sorted customer list: ['Alice', 'Ben', 'Charlie', 'Eve', 'Frank']
```

```
#Task_4:
```

```
grades = [85, 78, 92, 45, 33, 67, 88, 41]
```

```
def categorize_grades(grades):
    for score in grades:
        if score > 80:
            grade = 'A'
        elif score > 60:
            grade = 'B'
        elif score > 40:
            grade = 'C'
        else:
            grade = 'F'
        print(f"Score: {score} - Grade: {grade}")

def boost_grades(grades):
    boosted = list(map(lambda x: round(x * 1.05, 2), grades)) # Apply 5% boost to each grade
    return boosted
```

```
def boosted_above_90(boosted_grades):
    return list(filter(lambda x: x > 90, boosted_grades))
```

```
print("Grade Categories:")
categorize_grades(grades)
```

```
boosted_grades = boost_grades(grades)
print("\nBoosted Grades:")
print(boosted_grades)
```

```
grades_above_90 = boosted_above_90(boosted_grades)
print("\nBoosted Grades Above 90:")
print(grades_above_90)
```

```
→ Grade Categories:
Score: 85 - Grade: A
Score: 78 - Grade: B
Score: 92 - Grade: A
Score: 45 - Grade: C
Score: 33 - Grade: F
Score: 67 - Grade: B
Score: 88 - Grade: A
Score: 41 - Grade: C

Boosted Grades:
[89.25, 81.9, 96.6, 47.25, 34.65, 70.35, 92.4, 43.05]

Boosted Grades Above 90:
[96.6, 92.4]
```

```
#Task_5:
```

```
books = (
    ("To Kill a Mockingbird", "Harper Lee", 1960),
    ("1984", "George Orwell", 1949),
    ("The Great Gatsby", "F. Scott Fitzgerald", 1925)
)
tags = {"classic", "dystopian", "novel", "literature"}

second_book_author = books[1][1]
print(f"The author of the second book is: {second_book_author}")

new_book = ("Brave New World", "Aldous Huxley", 1932)
books = books + (new_book,)
print("\nUpdated books list after adding 'Brave New World':")
for book in books:
    print(book)
```

```
title, author, year = books[2]
print(f"\nDetails of the third book:\nTitle: {title}, Author: {author}, Year: {year}")

print("\nBook titles in the collection:")
for book in books:
    print(book[0])
```

```
tags.add("sci-fi")
print("\nUpdated tags set after adding 'sci-fi':", tags)

tags.discard("novel")
print("\nUpdated tags set after removing 'novel':", tags)
```



The author of the second book is: George Orwell

```
Updated books list after adding 'Brave New World':
('To Kill a Mockingbird', 'Harper Lee', 1960)
('1984', 'George Orwell', 1949)
('The Great Gatsby', 'F. Scott Fitzgerald', 1925)
('Brave New World', 'Aldous Huxley', 1932)
```

Details of the third book:

Title: The Great Gatsby, Author: F. Scott Fitzgerald, Year: 1925

Book titles in the collection:

To Kill a Mockingbird

1984

The Great Gatsby

Brave New World

Updated tags set after adding 'sci-fi': {'dystopian', 'sci-fi', 'novel', 'literature', 'classic'}

Updated tags set after removing 'novel': {'dystopian', 'sci-fi', 'literature', 'classic'}