Programming Fundamentals – Concept Practice Assignment

Question 1: Scholarship Eligibility System:
A university wants to identify students eligible for merit-based scholarships. Each student record contains roll number, name, and marks in three assessments.

[1]:
```
students_info = [
    {"Roll" :1 ,"Name" :"Ayesha", "marks" :[85,90,95]},
    {"Roll" :2 ,"Name" :"Ali", "marks" :[85,80,95]},
    {"Roll" :3 ,"Name" :"Akram", "marks" :[85,60,75]},
    {"Roll" :4 ,"Name" :"Ahsan", "marks" :[85,70,80]},
    {"Roll" :5 ,"Name" :"Adnan", "marks" :[85,88,95]},
    {"Roll" :6 ,"Name" :"Aiza", "marks" :[85,89,95]}
]
student_passed = [
    student for student in students_info
        if all(mark>=85 for mark in student["marks"])

]
print("Students who passed are:")
for a in student_passed:
    print(a["Name"])
```

```
Students who passed are:
Ayesha
Adnan
Aiza
```

[2]:
```
unique_name = {student["Name"] for student in students_info}
print(unique_name)
```

```
{'Aiza', 'Ali', 'Ayesha', 'Ahsan', 'Akram', 'Adnan'}
```

[3]:
```
def calculate_average(student):
    return sum(student["marks"]) / len(student["marks"])

students_info = [
    {"Roll" :1 ,"Name" :"Ayesha", "marks" :[85,90,95]},
    {"Roll" :2 ,"Name" :"Ali", "marks" :[85,80,95]},
    {"Roll" :3 ,"Name" :"Akram", "marks" :[85,60,75]},
    {"Roll" :4 ,"Name" :"Ahsan", "marks" :[85,70,80]},
    {"Roll" :5 ,"Name" :"Adnan", "marks" :[85,88,95]},
    {"Roll" :6 ,"Name" :"Aiza", "marks" :[85,89,95]}
]

for s in students_info:
    avg = calculate_average(s)
    print(s["Name"], avg)
```

```
Ayesha 90.0
Ali 86.66666666666667
Akram 73.33333333333333
Ahsan 78.33333333333333
Adnan 89.33333333333333
Aiza 89.66666666666667
```

[4]:
```
def scholarship(students_info):
    for student in students_info:
        avg = calculate_average(student)
        if avg >= 90:
            print(f"{student['Name']}- Average:{avg:.2f}")
scholarship(students_info)
```

```
Ayesha- Average:90.00
```

[5]:
```
def best_student(students_info):
    best = max(students_info, key=calculate_average)
    print("\nBest Performing Student:")
    print(f"{best['Name']} with Average: {calculate_average(best):.2f}")
best_student(students_info)
```

```

Best Performing Student:
Ayesha with Average: 90.00
```

Question 2: Warehouse Stock Monitoring

A warehouse records product movement during the day. Each record contains product name, units dispatched, and cost per unit.

[6]:
```
warehouse_records = [
    {"product": "Widget A", "units": 10, "cost": 5.0},
    {"product": "Widget B", "units": 5, "cost": 20.0},
    {"product": "Widget A", "units": 15, "cost": 4.5}
]

product_stats = {}
for rec in warehouse_records:
    p = rec['product']
    if p not in product_stats:
        product_stats[p] = {'units': 0, 'value': 0, 'costs': []}
        product_stats[p]['units'] += rec['units']
        product_stats[p]['value'] += rec['units'] * rec['cost']
        product_stats[p]['costs'].append(rec['cost'])

for p, stats in product_stats.items():
    print(f"Product: {p} | Total Units: {stats['units']} | Total Value: ${stats['value']:.2f}")
```

```
Product: Widget A | Total Units: 10 | Total Value: $50.00
Product: Widget B | Total Units: 5 | Total Value: $100.00
```

[7]:
```
max_prod = max(product_stats, key=lambda p: product_stats[p]['units'])
print(f"Max Dispatch Product: {max_prod}")
```

```
Max Dispatch Product: Widget A
```

[8]:
```
for p, stats in product_stats.items():
    avg_cost = sum(stats['costs']) / len(stats['costs'])
    print(f"Average Unit Cost for {p}: ${avg_cost:.2f}")
```

```
Average Unit Cost for Widget A: $5.00
Average Unit Cost for Widget B: $20.00
```

Question 3: Employee Performance Review

A company evaluates employee performance using quarterly ratings.

```
employees = [
    {"name": "Jamshed", "ratings": [3.0, 3.5, 4.0, 4.5]},
    {"name": "Ali", "ratings": [4.5, 4.0, 4.8, 5.0]},
    {"name": "Hamid", "ratings": [4.0, 4.2, 4.5, 4.6]}
]

improvers = [e['name'] for e in employees if all(e['ratings'][i] > e['ratings'][i-1] for i in range(1, len(e['ratings'])))]
print(f"Improved every quarter: {improvers}")
```

```
Improved every quarter: ['Jamshed', 'Hamid']
```

```
emp_names = {e['name'] for e in employees}
```

```
def calc_avg_perf(rating):
    return sum(rating['ratings']) / len(rating['ratings'])
employees = [
    {"name": "Jamshed", "ratings": [3.0, 3.5, 4.0, 4.5]},
    {"name": "Ali", "ratings": [4.5, 4.0, 4.8, 5.0]},
    {"name": "Hamid", "ratings": [4.0, 4.2, 4.5, 4.6]}
]
for s in employees:
    result = calc_avg_perf(s)
    print(s['ratings'], "average of rating is :" ,result)
```

```
[3.0, 3.5, 4.0, 4.5] average of rating is : 3.75
[4.5, 4.0, 4.8, 5.0] average of rating is : 4.575
[4.0, 4.2, 4.5, 4.6] average of rating is : 4.325
```

```
def calc_avg_perf(rating_list):
    return sum(rating_list) / len(rating_list)

def identify_high_performers(emp_list):
        return [e['name'] for e in emp_list if calc_avg_perf(e['ratings']) >= 4.5]
employees = [
    {"name": "Jamshed", "ratings": [3.0, 3.5, 4.0, 4.5]},
    {"name": "Ali", "ratings": [4.5, 4.0, 4.8, 5.0]},
    {"name": "Hamid", "ratings": [4.0, 4.2, 4.5, 4.6]}
]

result = identify_high_performers(employees)
print(f"High Performers (>= 4.5): {result}")
```

```
High Performers (>= 4.5): ['Ali']
```

```
def find_top_employee(emp_list):
    top = max(emp_list, key=lambda e: calc_avg_perf(e['ratings']))
    return top['name']
employees = [
    {"name": "Jamshed", "ratings": [3.0, 3.5, 4.0, 4.5]},
    {"name": "Ali", "ratings": [4.5, 4.0, 4.8, 5.0]},
    {"name": "Hamid", "ratings": [4.0, 4.2, 4.5, 4.6]}
]
top_emp = find_top_employee(employees)
print(f"Top Rated Employee: {top_emp}")
```

```
Top Rated Employee: Ali
```

Question 4: Online Course Sales Summary

An e-learning platform tracks course purchases.

```
courses = [
    {"name": "Python for Beginners",     "enrollments": 450, "price": 99},
    {"name": "Web Development Bootcamp", "enrollments": 320, "price": 149},
    {"name": "Data Science Mastery",     "enrollments": 580, "price": 199},
    {"name": "Machine Learning A-Z",     "enrollments": 410, "price": 179},
    {"name": "Java Programming",         "enrollments": 290, "price": 129}
]
for course in courses:
    print(f"{course['name']}: {course['enrollments']} students")
```

```
Python for Beginners: 450 students
Web Development Bootcamp: 320 students
Data Science Mastery: 580 students
Machine Learning A-Z: 410 students
Java Programming: 290 students
```

```
for course in courses:
    revenue = course['enrollments'] * course['price']
    print(f"{course['name']}: ${revenue}")
```

```
Python for Beginners: $44550
Web Development Bootcamp: $47680
Data Science Mastery: $115420
Machine Learning A-Z: $73390
Java Programming: $37410
```

```
most_popular = courses[0]
for course in courses:
    if course['enrollments'] > most_popular['enrollments']:
        most_popular = course
print("The most popular course:")
print(f"   Course: {most_popular['name']}")
print(f"   Enrollments: {most_popular['enrollments']} students")
```

```
3. The most popular course:
   Course: Data Science Mastery
   Enrollments: 580 students
```

```
print("Course fee for each course:")
for course in courses:
    print(f"{course['name']}: ${course['price']}")
```

```
Course fee for each course:
Python for Beginners: $99
Web Development Bootcamp: $149
Data Science Mastery: $199
Machine Learning A-Z: $179
Java Programming: $129
```