

---

# System Requirements Specification Index

For

Python Basics and NumPy, Pandas

Usecase 3

1.0



### **Use Case:1 Car Inventory Management (carinventory.py)**

#### **1) Write a Python program to search for cars within a given budget.**

- Define a function `search_by_budget(inventory, max_price)`.
- The function should:
  - Filter and display cars where the price is less than or equal to `max_price`.
  - If no cars match the criteria, print an appropriate message.
  - Return the filtered list of cars.

#### **2) Write a Python program to save the car inventory into a JSON file.**

- Define a function `save_inventory(inventory, filename)`.
- The function should:
  - Convert the car inventory into JSON format.
  - Save it to a file named `car_inventory.json`.
  - Print a success message after saving.
  - Return the filename.

#### **3) Write a Python program to execute all inventory operations.**

- Implement a `main()` function that:
  - Calls `display_cars(car_inventory)` to show all available cars.
  - Calls `search_by_budget(car_inventory, 25000)` to find cars under \$25,000.
  - Calls `save_inventory(car_inventory)` to store inventory data in a JSON file.

### **Use Case2: Student Management System (StudentCourseManagement.py)**

#### **1) Write a Python program to store and display a list of student names.**

- Define a function `student_names()`.
- The function should:
  - Store a list of students: "John", "Emma", "Sophia", "Michael", "Daniel".
  - Append "Olivia" to the list.
  - Return the updated student list.

#### **2) Write a Python program to store student course enrollments using a dictionary.**

- Define a function `student_courses()`.
- The function should:
  - Store student names as keys and their enrolled courses as tuple values.
  - Add a new entry for "Olivia" with courses ("Biology", "History").
  - Return the updated dictionary.

#### **3) Write a Python program to store and display unique subjects across all students.**

- Define a function `unique_subjects()`.
- The function should:
  - Use a set to store unique subjects across all student enrollments.
  - Ensure duplicate subjects (e.g., "Math") appear only once.
  - Add "Economics" as a new subject.
  - Return the updated set of unique subjects.

### **Use Case3: Student Marks Analysis (StudentMarksAnalysis.py)**

#### **1) Write a Python program to compute basic statistics for student marks.**

- Define a function `analyze_marks(marks)`.
- The function should:
  - Compute the average, maximum, and minimum marks using NumPy.
  - Return these three statistics.

#### **2) Write a Python program to classify students based on their marks.**

- Define a function `classify_grades(marks)`.
- The function should:

- Assign grades based on the following criteria:
    - A: mark  $\geq 90$
    - B: mark  $\geq 80$
    - C: mark  $\geq 70$
    - D: mark  $< 70$
  - Return the list of grades.
- 3) Given a list of student marks, analyze and display the results.
  - Use NumPy to store student marks.
  - Call analyze\_marks() to get statistical insights.
  - Call classify\_grades() to determine student grades.
  - Display:
    - The list of marks
    - Average marks (rounded to 2 decimal places)
    - Highest and lowest marks
    - Corresponding grades

### **Execution Steps to Follow:**

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
3. This editor Auto Saves the code
4. If you want to exit (logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
5. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.

6. To setup environment:

You can run the application without importing any packages

7. To launch application:

**Python3 carinventory.py**

**Python3 StudentMarksAnalysis.py**

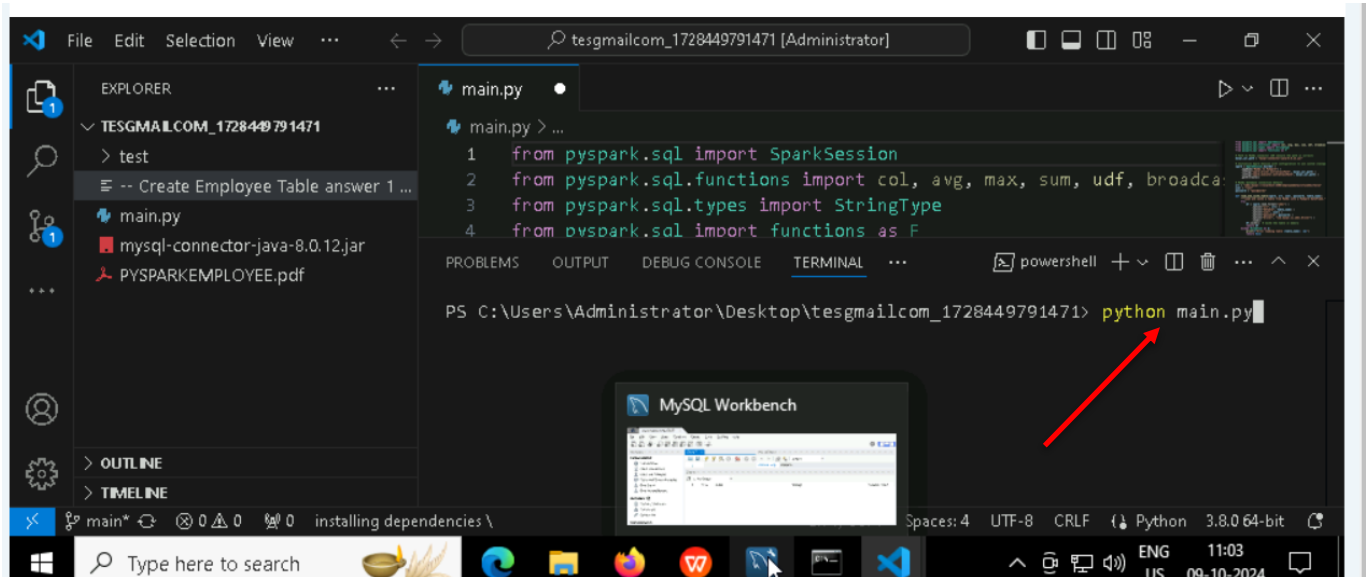
**Python3 StudentMarksAnalysis.py**

To run Test cases:

**Python3 -m unittest**

Before Final Submission also, you need to use CTRL+Shift+B-command compulsorily on code IDE.  
This will push or save the updated contents in the internalgit/repository for code

### Screen shot to run the program

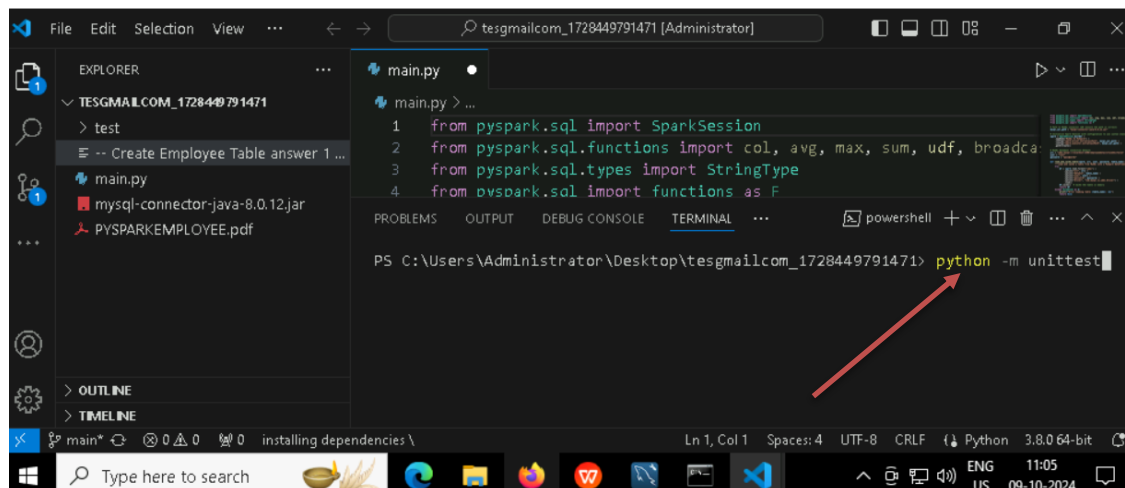


To run the application

Python3 carinventory.py

Python3 StudentMarksAnalysis.py

Python3 StudentMarksAnalysis.py



To run the testcase

- Python3 -m unittest

## Screenshot to push the application to github

-----X-----

You can run test cases as many numbers of times and at any stage of Development, to check howmany test cases are passed/failed and accordingly refactor your code.

1. Make sure before final submission you commit all changes to git. For that

In the terminal use the command git status

```
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$
```

a. git add .

```
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$ git add .
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$
```

b. git commit -m "First commit"  
(You can provide any message every time you commit)

```
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$ git commit -m "firstcommit"
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$
```

c. git push

```
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$ git push
Everything up-to-date
coder@cddqw20250224t0114391hh3bxmh:/home/myproject/cddqwgmailcom_20250224T011439$
```

After you have pushed your code Finally click on the final submission button


Python-Level1-Template

Time Remaining: 02:54:04

Submit Assessment

Project 1

1 Python-Level1-Template



URL  
https://compiler.techademy.com/vsc/cddqw20250224t0114391...

Password  
mA4DpEBElF

Kindly use this password to login into the IDE  
All the questions, files and related documents for this assessment will be available within the IDE

fighq fdsds  
cddqw@gmail.com  
Speed Test: >= 10Mbps Live: 0.41Mbps

0564 techademy@iit.com Privacy Policy

All Rights Reserved by Techademy Learning Solutions Private Limited

Click on the submit assessment button after you have pushed the code

-----X-----