CS6103E Software Systems Laboratory

Monsoon Semester (2025-26)

1st Year M.Tech CSE/CS(IS)

Python Programming

The objective is to learn the following:

- Basic data structures like lists, dictionary and tuples.
- Fundamental Programming concepts like loops and conditional logic.
- Creating user defined functions and handling csv and text files.
- Working with pandas.

Submission date: On or before 13.11.2025 (11:59 pm)
Mode of submission: as a single `.py` file named as per the format `P1_<FIRST NAME>_<ROLLNO>.py` (e.g. `P1_saritha_m240064CS.py`)

1. The *tested.csv* file consists of passenger details from the Titanic dataset. The dataset contains information such as **passenger class, name, gender, age, ticket number, fare, and port of embarkation**. Read the file into a dataframe named as 'titanic' and perform the following operations:

i. Display the shape of the dataset (number of rows and columns) and list all column names with their corresponding data types.

ii. Read the dataset into a pandas dataframe and identify the columns containing missing values. If the number of missing values in a column is more than 50%, then delete that column. Remove all rows that contain any missing value from the dataframe.
Display the number of rows and columns before and after removing missing values. Save the cleaned dataframe as a new CSV file named *cleaned.csv*.

iii. The dataset *tested.csv* contains a few missing values in attributes like Age, Fare, and Cabin. Replace the missing values in Age and Fare columns with their respective mean values, and replace missing values in Cabin with the string 'Unknown'.

2. Consider the same dataset *tested.csv* and perform the following operations:

i. The `Name` field in the dataset is stored in the format `"LastName, Title. FirstName"`. Modify all names so that they are displayed in the format `"Title FirstName LastName"`.

Store the reformatted names in a new column named **Formatted Name** and display the first ten records of the dataframe after modification.

Example:

Original Name (in dataset)	**Expected Output (Formatted Name)**
Smith, Mr. John	Mr. John Smith
Taylor, Mrs. Anna	Mrs. Anna Taylor

ii. Create a **unique ID** for each passenger starting with `'P'`, followed by the **passenger number** which corresponds to the row number in the dataset, and finally append the passenger's **class (Pclass)** to the ID using an underscore (`'_'`).

For example, if a passenger is present in the 123^{rd} row of the dataset with class as 3, his/her ID should be `P123_3`. Note that the row number starts from 0, but we will consider it as 1.

Create a dictionary where the **key** should be the unique ID of the passenger, and the **value** should consist of a list containing the passenger's `Name`, `Pclass`, and `Fare`.

Example:

```
{
  P1_3: ['Kelly, Mr. James', 3, 7.8292]
}
```

After creating the dictionary, display the **top 5 passengers who paid the highest fare**, showing their `Name`, `Pclass`, and `Fare`.

Example Output:

```
Name                                Pclass    Fare
Allen, Miss. Elisabeth Walton          1    512.3292
Ryerson, Miss. Emily Borie             1    262.3750
...
```

3. The shared text file *data.txt* contains a mixture of letters, symbols, numbers, and encoded words. The file is not formatted properly — it includes random characters,

hidden codes, and mixed case words. You are required to process this text file to extract meaningful information.

Example contents of `data.txt`:

```
abC#12@world2025$!!HELLO#99good_morning43^#Bye007%done$
```

Perform the following operations using Python:

i. Read the contents of `data.txt` into a string variable and write a Python function to extract all **numeric codes** (continuous sequences of digits) present in the text.

Example Output:

```
Codes: [12, 2025, 99, 43, 007]
```

ii. From the same text, extract and display all **words that end with a number**.

Example Output:

```
Words with numbers at end: world2025 good_morning43 Bye007
```

iii. Identify and print all **uppercase words** present in the text.

Example Output:

```
Uppercase words: HELLO
```

iv. Replace all special characters with a single space, and print the **cleaned** text.

Example Output:

```
Cleaned text: abC 12 world2025  HELLO 99 good_morning43  Bye007 done
```