

**Data Technician**

|  |
| --- |
|  |

|  |
| --- |
| Name: Michael Thirlaway |
| Course Date: 14th 07 2025 |
|  |

**Table of contents**

[Day 2: Task 1 3](#_Toc1765240926)

[Day 3: Task 1 3](#_Toc873857632)

[Exercise 1: Loading and Exploring the Data 4](#_Toc1363711086)

[Exercise 2: Indexing and Slicing 4](#_Toc1679584900)

[Exercise 3: Data Manipulation 4](#_Toc998449442)

[Exercise 4: Aggregation and Grouping 5](#_Toc267294297)

[Exercise 5: Advanced Operations 5](#_Toc1232452443)

[Exercise 6: Exporting Data 6](#_Toc1251390377)

[Exercise 7: If finished early try visualising the results 6](#_Toc66584577)

[Day 4: Task 1 7](#_Toc1452452084)

[Day 4: Task 2 8](#_Toc934594145)

[Course Notes 8](#_Toc1227711066)

[Additional Information 9](#_Toc1836543406)

# Day 2: Task 1

It is a common software development interview question to create the below with a certain programming language. Create the below using Python syntax, test it and past the completed syntax and output below.

FizzBuzz:

Go through the integers from 1 to 100.

If a number is divisible by 3, print "fizz."

If a number is divisible by 5, print "buzz."

If a number is both divisible by 3 and by 5, print "fizzbuzz."

Otherwise, print just the number.

|  |  |
| --- | --- |
| Paste your completed work to the right | for number in range(1, 101): # go through numbers 1 to 100  if number % 3 == 0 and number % 5 == 0:  print("fizzbuzz")  elif number % 3 == 0:  print("fizz")  elif number % 5 == 0:  print("buzz")  else:  print(number)  43  44  fizzbuzz  46  47  fizz  49  buzz  fizz  52  53  fizz  buzz  56  fizz  58  59  fizzbuzz  61  62  fizz  64  buzz  fizz  67  68  fizz  buzz  71  fizz  73  74  fizzbuzz  76  77  fizz  79  buzz  fizz  82  83  fizz  buzz  86  fizz  88  89  fizzbuzz  91  92  fizz  94  buzz  fizz  97  98  fizz  buzz |

# **Day 3: Task 1**

Download the ‘student.csv’, complete the below exercises as a group and paste your input and output. Although this is a group activity, everyone should have the below answered so it supports your portfolio:

### **Exercise 1: Loading and Exploring the Data**

1. Question: "Write the code to read a CSV file into a Pandas DataFrame."
2. Question: "Write the code to display the first 5 rows of the DataFrame."
3. Question: "Write the code to get the information about the DataFrame."
4. Question: "Write the code to get summary statistics for the DataFrame."

|  |
| --- |
| 1 Read a CSV file into a Pandas DataFrame  # (Change the filename here if yours is different)  csv\_path = "student.csv"  df\_students = pd.read\_csv(csv\_path)  print("1) Loaded DataFrame from CSV.")  print("Shape (rows, columns):", df\_students.shape, "\n")  2) Display the first 5 rows of the DataFrame  first\_five = df\_students.head() # save to a variable to keep it clear  print("2) First 5 rows of the DataFrame:")  print(first\_five, "\n")  3 Get information about the DataFrame (column names, types, nulls, etc.)  print("3) DataFrame info:")  info\_result = df\_students.info() # .info() already prints; capturing return just to be explicit  print(info\_result, "\n") # will print 'None' after the info block (that’s normal)  4 Get summary statistics for the DataFrame  # Usually .describe() shows numeric columns by default.  stats\_numeric = df\_students.describe()  print("4) Summary statistics (numeric columns):")  print(stats\_numeric, "\n") |

### **Exercise 2: Indexing and Slicing**

1. Question: "Write the code to select the 'name' column."
2. Question: "Write the code to select the 'name' and 'mark' columns."
3. Question: "Write the code to select the first 3 rows."
4. Question: "Write the code to select all rows where the 'class' is 'Four'."

|  |
| --- |
| df\_students = pd.read\_csv("student.csv")  1 Select the 'name' column  names\_only = df\_students['name']  print("1) The 'name' column:")  print(names\_only.head(), "\n") # just show first few for readability  2 Select the 'name' and 'mark' columns  names\_and\_marks = df\_students[['name', 'mark']]  print("2) The 'name' and 'mark' columns:")  print(names\_and\_marks.head(), "\n")  3 Select the first 3 rows  first\_three\_rows = df\_students.head(3)  print("3) The first 3 rows of the DataFrame:")  print(first\_three\_rows, "\n")  4 Select all rows where the 'class' is 'Four'  class\_four = df\_students[df\_students['class'] == 'Four']  print("4) All rows where class is 'Four':")  print(class\_four.head(), "\n") # again, head() to keep it neat |

### **Exercise 3: Data Manipulation**

1. Question: "Write the code to add a new column 'passed' that indicates whether the student passed (mark >= 60)."
2. Question: "Write the code to rename the 'mark' column to 'score'."
3. Question: "Write the code to drop the 'passed' column."

|  |
| --- |
| df\_students = pd.read\_csv("student.csv")  1 Add a new column 'passed' (True if mark >= 60, else False)  df\_students['passed'] = df\_students['mark'] >= 60  print("1) DataFrame with new 'passed' column:")  print(df\_students.head(), "\n")  2 Rename the 'mark' column to 'score'  df\_students = df\_students.rename(columns={'mark': 'score'})  print("2) DataFrame with 'mark' column renamed to 'score':")  print(df\_students.head(), "\n")  3 Drop the 'passed' column  df\_students = df\_students.drop(columns=['passed'])  print("3) DataFrame after dropping the 'passed' column:")  print(df\_students.head(), "\n") |

### **Exercise 4: Aggregation and Grouping**

1. Question: "Write the code to group the DataFrame by the 'class' column and calculate the mean 'mark' for each group."
2. Question: "Write the code to count the number of students in each class."
3. Question: "Write the code to calculate the average mark for each gender."

|  |
| --- |
| df\_students = pd.read\_csv("student.csv")  1 Group by 'class' and calculate the mean 'mark'  mean\_mark\_by\_class = df\_students.groupby('class')['mark'].mean()  print("1) Mean mark for each class:")  print(mean\_mark\_by\_class, "\n")  2 Count the number of students in each class  count\_by\_class = df\_students.groupby('class')['name'].count()  print("2) Number of students in each class:")  print(count\_by\_class, "\n")  3 Calculate the average mark for each gender  mean\_mark\_by\_gender = df\_students.groupby('gender')['mark'].mean()  print("3) Average mark for each gender:")  print(mean\_mark\_by\_gender, "\n") |

### **Exercise 5: Advanced Operations**

1. Question: "Write the code to create a pivot table with 'class' as rows, 'gender' as columns, and 'mark' as values."
2. Question: "Write the code to create a new column 'grade' where marks >= 85 are 'A', 70-84 are 'B', 60-69 are 'C', and below 60 are 'D'."
3. Question: "Write the code to sort the DataFrame by 'mark' in descending order."

|  |
| --- |
| df\_students = pd.read\_csv("student.csv")  1 Create a pivot table with 'class' as rows, 'gender' as columns, and 'mark' as values  pivot\_table = df\_students.pivot\_table(values="mark", index="class", columns="gender", aggfunc="mean")  print("1) Pivot table of mean mark by class and gender:")  print(pivot\_table, "\n")  2 Create a new column 'grade' using mark ranges  def assign\_grade(m):  if m >= 85:  return "A"  elif m >= 70:  return "B"  elif m >= 60:  return "C"  else:  return "D"  df\_students['grade'] = df\_students['mark'].apply(assign\_grade)  print("2) DataFrame with new 'grade' column:")  print(df\_students[['name', 'mark', 'grade']].head(), "\n")  3 Sort the DataFrame by 'mark' in descending order  sorted\_df = df\_students.sort\_values(by='mark', ascending=False)  print("3) DataFrame sorted by mark (descending):")  print(sorted\_df[['name', 'mark', 'grade']].head(), "\n") |

### **Exercise 6: Exporting Data**

1. Question: "Write the code to save the DataFrame with the new 'grade' column to a new CSV file."

|  |
| --- |
| df\_students = pd.read\_csv("student.csv")  def assign\_grade(m):  if m >= 85:  return "A"  elif m >= 70:  return "B"  elif m >= 60:  return "C"  else:  return "D"  df\_students['grade'] = df\_students['mark'].apply(assign\_grade)  1 Save the DataFrame with the new 'grade' column to a new CSV file  df\_students.to\_csv("students\_with\_grades.csv", index=False)  print("DataFrame saved successfully as 'students\_with\_grades.csv'") |

### **Exercise 7: If finished early try visualising the results**

|  |
| --- |
|  |

# **Day 4: Task 1**

Using the ‘GDP (nominal) per Capita.csv’ which can be downloaded from the shared Folder, complete the below exercises and paste your input and output. Work individually, but we will work and support each other in the room.

* Read and save the ‘GDP (nominal) per Capita’ data to a data frame called “df” in Jyputer notebook
* Print the first 10 rows
* Print the last 5 rows
* Print ‘Country/Territory’ and ‘UN\_Region’ columns

|  |
| --- |
| *First, I imported pandas before proceeding with the task:* import pandas as pd  df = pd.read\_csv("GDP (nominal) per Capita.csv",encoding= 'unicode\_escape',  index\_col=0) |

# **Day 4: Task 2**

Back with ‘GDP (nominal) per Capita’. As a group, import and work your way through the Day\_4\_Python\_Activity.ipynb notebook which can be found on the shared Folder. There are questions to answer, but also opportunities to have fun with the data – paste your input and output below.

Once complete, and again as a group, work with some more data and have some fun –there is no set agenda for this section, other than to embed the skills developed this week. Paste your input and output below and upon return we’ll discuss progress made.

[Additional data found here.](https://justit831-my.sharepoint.com/:f:/g/personal/danpe_justit_co_uk/Er0ybU9i0AZKiuGaCWZyj2ABoqKD23zwLGdJf3WlaixpRA?e=QVj2Bs)

|  |
| --- |
|  |

|  |
| --- |
| **Course Notes** |

It is recommended to take notes from the course, use the space below to do so, or use the revision guide shared with the class:

|  |
| --- |
| **Additional Information** |

We have included a range of additional links to further resources and information that you may find useful, these can be found within your revision guide.

**END OF WORKBOOK**

**Please check through your work thoroughly before submitting and update the table of contents if required.**

**Please send your completed work booklet to your trainer.**