

Solutions

Solution 1

```
a = int(input("Enter first integer: "))
b = int(input("Enter second integer: "))
print(f"Sum: {a + b}")
print(f"Product: {a * b}")
try:
    print(f"Division: {a / b}")
except ZeroDivisionError:
    print("Division by zero is undefined.")
```

Solution 2

```
sentence = input("Enter a sentence: ")
vowels = "aeiouAEIOU"
count = sum(1 for char in sentence if char in vowels)
print(f"Number of vowels: {count}")
```

Solution 3

```
with open("text_file.txt", "r") as file:
    text = file.read()
    word_count = len(text.split())
print(f"Word count: {word_count}")
```

Solution 4

```
import random

with open("random_numbers.txt", "a") as file:
    for _ in range(5):
        file.write(f"{random.randint(1, 100)}\n")
```

Solution 5

```
import csv

with open("scores.csv", "r") as file:
    reader = csv.DictReader(file)
    scores = [(row["name"], int(row["score"])) for row in reader]
    avg_score = sum(score for _, score in scores) / len(scores)
    above_avg = [name for name, score in scores if score > avg_score]
    print("Names with scores above average:", above_avg)
```

Solution 6

```
import csv

name = input("Enter name: ")
balance = float(input("Enter balance: "))
with open("balance_data.csv", "a", newline="") as file:
```

```
writer = csv.writer(file)
writer.writerow([name, balance])
```

Solution 7

```
import pandas as pd

df = pd.read_csv("data.csv")
df = df.applymap(lambda x: x * 1.1 if isinstance(x, (int, float)) else x)
df.to_csv("modified_data.csv", index=False)
```

Solution 8

```
import pandas as pd

file_path = "multi_sheet_data.xlsx"
output_path = "averages_output.xlsx"
with pd.ExcelWriter(output_path) as writer:
    for sheet_name in pd.ExcelFile(file_path).sheet_names:
        df = pd.read_excel(file_path, sheet_name=sheet_name)
        avg = df.mean().to_frame("Average").T
        avg.to_excel(writer, sheet_name=sheet_name)
```

Solution 9

```
import pandas as pd

file_path = "data.xlsx"
df = pd.read_excel(file_path, sheet_name="Sheet1")
threshold = float(input("Enter threshold: "))
filtered_df = df[df["Column_of_interest"] > threshold]
with pd.ExcelWriter(file_path, mode="a") as writer:
    filtered_df.to_excel(writer, sheet_name="Filtered Data")
```

Solution 10

```
import pandas as pd

data = {
    "Name": ["Alice", "Bob", "Charlie"],
    "Age": [25, 30, 35],
    "Grade": ["A", "B", "A"]
}
df = pd.DataFrame(data)
df.to_csv("output_data.csv", index=False)
df.to_excel("output_data.xlsx", index=False)
```

Checkpoint

```
In [ ]: # Load the generated large CSV file
df = pd.read_csv(file_name)

# Step 1: Filter out all rows where the stock price is greater than $2000
filtered_df = df[df['Price'] > 2000]
```

```
# Step 2: Add a new column called 'Value', calculated as Price * Volume
df['Value'] = df['Price'] * df['Volume']

# Step 3: Group the data by 'Stock' and calculate the average price for each stock
avg_price_per_stock = df.groupby('Stock')['Price'].mean()

# Save the manipulated DataFrame to a new CSV file
df.to_csv('/modified_'+file_name, index=False)

# Display outputs for verification
filtered_df.head(), df.head(), avg_price_per_stock
```