

Object-Oriented Programming

The assignments are Programming Exercises 4.3 on page 203, 5.5 on page 280, and 10.2 on page 547 of the textbook (Chapters 4, 5, 10). For your convenience, the exercises are shown below. Please copy here your source codes, including your input and output screenshots. Please upload this document along with your source files (i.e., the .py files) on Canvas by the due date.

4.3 (*Budget Analysis*) Write a program that asks the user to enter the amount that he or she has budgeted for a month. A loop should then prompt the user to enter each of his or her expenses for the month and keep a running total. When the loop finishes, the program should display the amount that the user is over or under the budget.

Input: def main():

```
    budget = float(input("Enter your monthly budget: $"))
```

```
    total_expenses = 0.0
```

```
    while True:
```

```
        expense = input("Enter an expense (or type 'done' to finish): ")
```

```
        if expense.lower() == 'done':
```

```
            break
```

```
        try:
```

```
            total_expenses += float(expense)
```

```
        except ValueError:
```

```
            print("Invalid input. Please enter a number.")
```

```
print("\nBudget Summary")
```

```
print(f"Budgeted amount: ${budget:.2f}")
```

```
print(f"Total expenses: ${total_expenses:.2f}")
```

```
difference = budget - total_expenses
```

```
if difference > 0:
```

```
    print(f"You are under budget by ${difference:.2f}.")
```

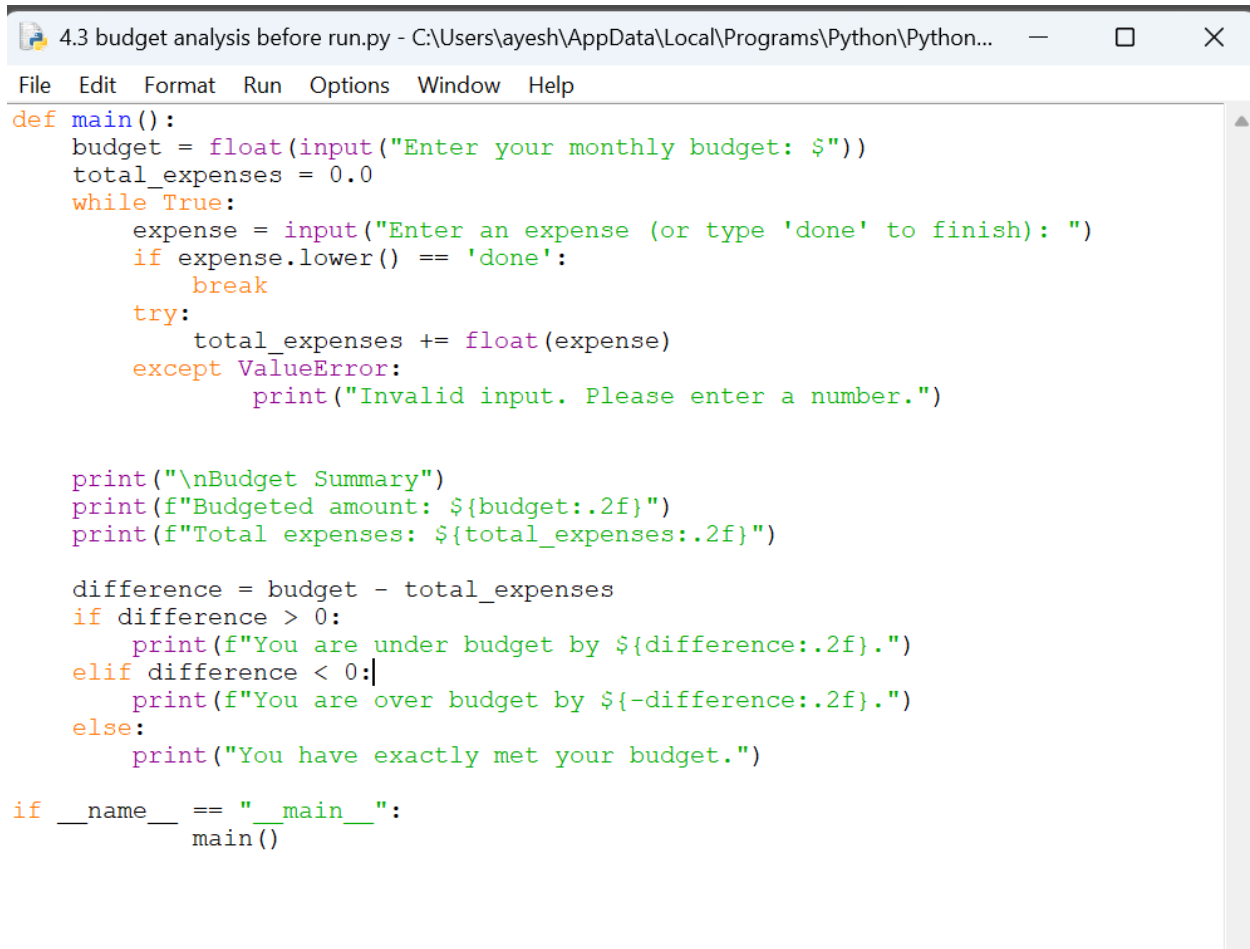
```
elif difference < 0:
```

```
    print(f"You are over budget by ${-difference:.2f}.")
```

```
else:
```

```
print("You have exactly met your budget.")
```

```
if __name__ == "__main__":  
    main()
```

A screenshot of a Python IDE window titled "4.3 budget analysis before run.py". The window shows a Python script for budget analysis. The script defines a main function that takes a monthly budget as input, then enters a loop to collect expenses. It uses try-except to handle invalid input. After collecting expenses, it calculates the difference between the budget and total expenses and prints a summary message. The script is wrapped in an if __name__ == "__main__": block.

```
4.3 budget analysis before run.py - C:\Users\ayesh\AppData\Local\Programs\Python\Python...  
File Edit Format Run Options Window Help  
def main():  
    budget = float(input("Enter your monthly budget: $"))  
    total_expenses = 0.0  
    while True:  
        expense = input("Enter an expense (or type 'done' to finish): ")  
        if expense.lower() == 'done':  
            break  
        try:  
            total_expenses += float(expense)  
        except ValueError:  
            print("Invalid input. Please enter a number.")  
  
    print("\nBudget Summary")  
    print(f"Budgeted amount: ${budget:.2f}")  
    print(f"Total expenses: ${total_expenses:.2f}")  
  
    difference = budget - total_expenses  
    if difference > 0:  
        print(f"You are under budget by ${difference:.2f}.")  
    elif difference < 0:  
        print(f"You are over budget by ${-difference:.2f}.")  
    else:  
        print("You have exactly met your budget.")  
  
if __name__ == "__main__":  
    main()
```

Output: Enter your monthly budget: \$400

Enter an expense (or type 'done' to finish): 20

Enter an expense (or type 'done' to finish): 35

Enter an expense (or type 'done' to finish): 49

Enter an expense (or type 'done' to finish): done

Budget Summary

Budgeted amount: \$400.00

Total expenses: \$104.00

You are under budget by \$296.00.

```
IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\ayesh\AppData\Local\Programs\Python\Python311\4.3 budget ana
lysis before run.py
Enter your monthly budget: $400
Enter an expense (or type 'done' to finish): 20
Enter an expense (or type 'done' to finish): 35
Enter an expense (or type 'done' to finish): 49
Enter an expense (or type 'done' to finish): done

Budget Summary
Budgeted amount: $400.00
Total expenses: $104.00
You are under budget by $296.00.
>>>
```

5.5 (*Property Tax*) A county collects property taxes on the assessment value of property, which is 60 percent of the property's actual value. For example, if an acre of land is valued at \$10,000, its assessment value is \$6,000. The property tax is then 72¢ for each \$100 of the assessment value. The tax for the acre assessed at \$6,000 will be \$43.20. Write a program that asks for the actual value of a piece of property and displays the assessment value and property tax.

Input: def main():

```
    actual_value = float(input("Enter the actual value of the property: $"))
```

```
    assessment_value = actual_value * 0.60
```

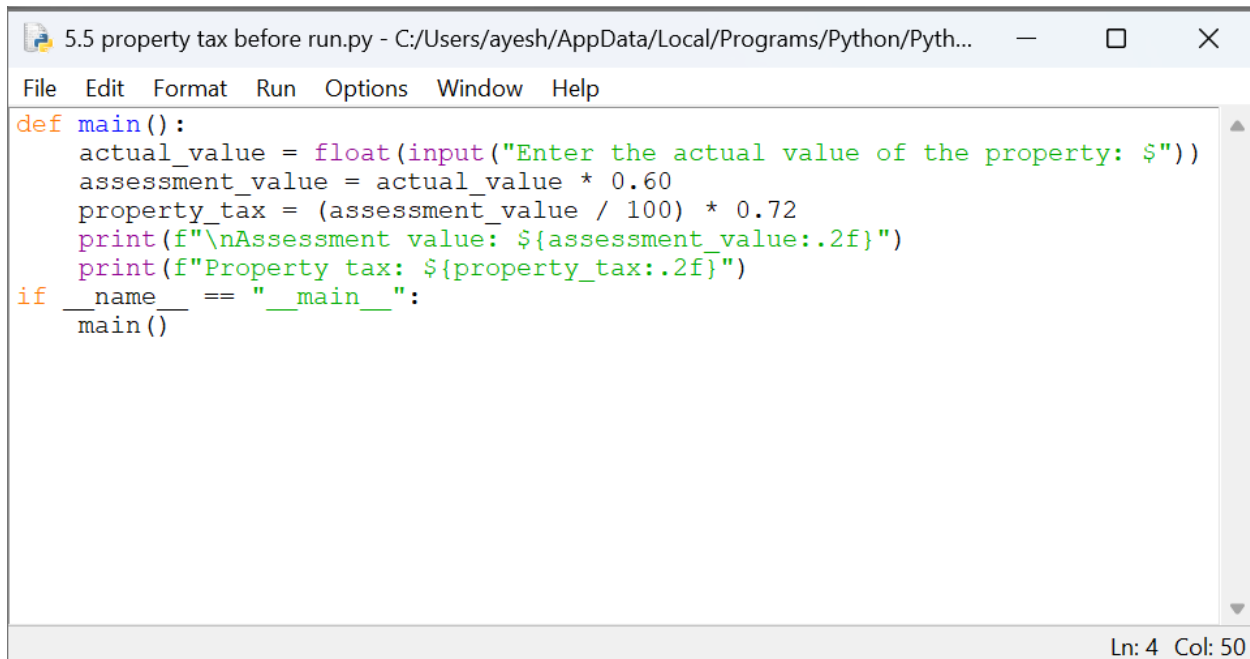
```
    property_tax = (assessment_value / 100) * 0.72
```

```
    print(f"\nAssessment value: ${assessment_value:.2f}")
```

```
    print(f"Property tax: ${property_tax:.2f}")
```

if __name__ == "__main__":

```
    main()
```

A screenshot of a Python IDE window titled "5.5 property tax before run.py - C:/Users/ayesh/AppData/Local/Programs/Python/Pyth...". The window contains the following Python code:

```
def main():
    actual_value = float(input("Enter the actual value of the property: $"))
    assessment_value = actual_value * 0.60
    property_tax = (assessment_value / 100) * 0.72
    print(f"\nAssessment value: ${assessment_value:.2f}")
    print(f"Property tax: ${property_tax:.2f}")
if __name__ == "__main__":
    main()
```

The status bar at the bottom right indicates "Ln: 4 Col: 50".

Output:

Enter the actual value of the property: \$380000

Assessment value: \$228000.00

Property tax: \$1641.60

```
IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/ayesh/AppData/Local/Programs/Python/Python311/5.5 property tax before run.py
Enter the actual value of the property: $380000

Assessment value: $228000.00
Property tax: $1641.60
>>> |
```

10.2 (*Car Class*) Write a class named Car that has the following data attributes:

- `__year_model` (for the car's year model)
- `__make` (for the make of the car)
- `__speed` (for the car's current speed)

The Car class should have an `__init__` method that accepts the car's year model and make as arguments. These values should be assigned to the object's `__year_model` and `__make` data attributes. It should also assign 0 to the `__speed` data attribute.

The class should also have the following methods:

- `accelerate`
The `accelerate` method should add 5 to the speed data attribute each time it is called.
- `brake`
The `brake` method should subtract 5 from the speed data attribute each time it is called.
- `get_speed`
The `get_speed` method should return the current speed.

Next, design a program that creates a Car object then calls the `accelerate` method five times. After each call to the `accelerate` method, get the current speed of the car and display it. Then call the `brake` method five times. After each call to the `brake` method, get the current speed of the car and display it.

Input:

class Car :

```
def __init__(self, make, model, year):
```

```
    self.__make = make
```

```
    self.__model = model
```

```
    self.__year = year
```

```
    self.__speed = 0
```

```
def display_info(self):
```

```
    print(f'Make: {self.__make}')
```

```
    print(f'Model: {self.__model}')
```

```
    print(f'Year: {self.__year}')
```

```
    print(f'Speed: {self.__speed}')
```

```
def accelerate(self):
```

```
    self.__speed += 5
```

```

def brake(self):
    if self.__speed >= 5:
        self.__speed -=5
    else:
        self.__speed = 0
def get_speed(self):
    return self.__speed

def main():
    my_car = Car("Toyota", "Corolla SE", 2017)
    print("Car Information:")
    my_car.display_info()
    print("\nAccelerating:")
    for i in range(5):
        my_car.accelerate()
        print(f'Current speed: {my_car.get_speed()}')
    print("\nBraking:")
    for i in range(5):
        my_car.brake()
        print(f'Current speed: {my_car.get_speed()}')

if __name__ == "__main__":
    main()

```

File Edit Format Run Options Window Help

```
class Car :
    def __init__(self, make, model, year):|
        self.__make = make
        self.__model = model
        self.__year = year
        self.__speed = 0

    def display_info(self):
        print(f"Make: {self.__make}")
        print(f"Model: {self.__model}")
        print(f"Year: {self.__year}")
        print(f"Speed: {self.__speed}")

    def accelerate(self):
        self.__speed += 5

    def brake(self):
        if self.__speed >= 5:
            self.__speed -=5
        else:
            self.__speed = 0
    def get_speed(self):
        return self.__speed

def main():
    my_car = Car("Toyota", "Corolla SE", 2017)
    print("Car Information:")
    my_car.display_info()
    print("\nAccelerating:")
    for i in range(5):
        my_car.accelerate()
        print(f"Current speed: {my_car.get_speed()}")
    print("\nBraking:")
    for i in range(5):
        my_car.brake()
        print(f"Current speed: {my_car.get_speed()}")

if __name__ == "__main__":
    main()
```

Output:

Car Information:

Make: Toyota

Model: Corolla SE

Year: 2017

Speed: 0

Accelerating:

Current speed: 5

Current speed: 10

Current speed: 15

Current speed: 20

Current speed: 25

Braking:

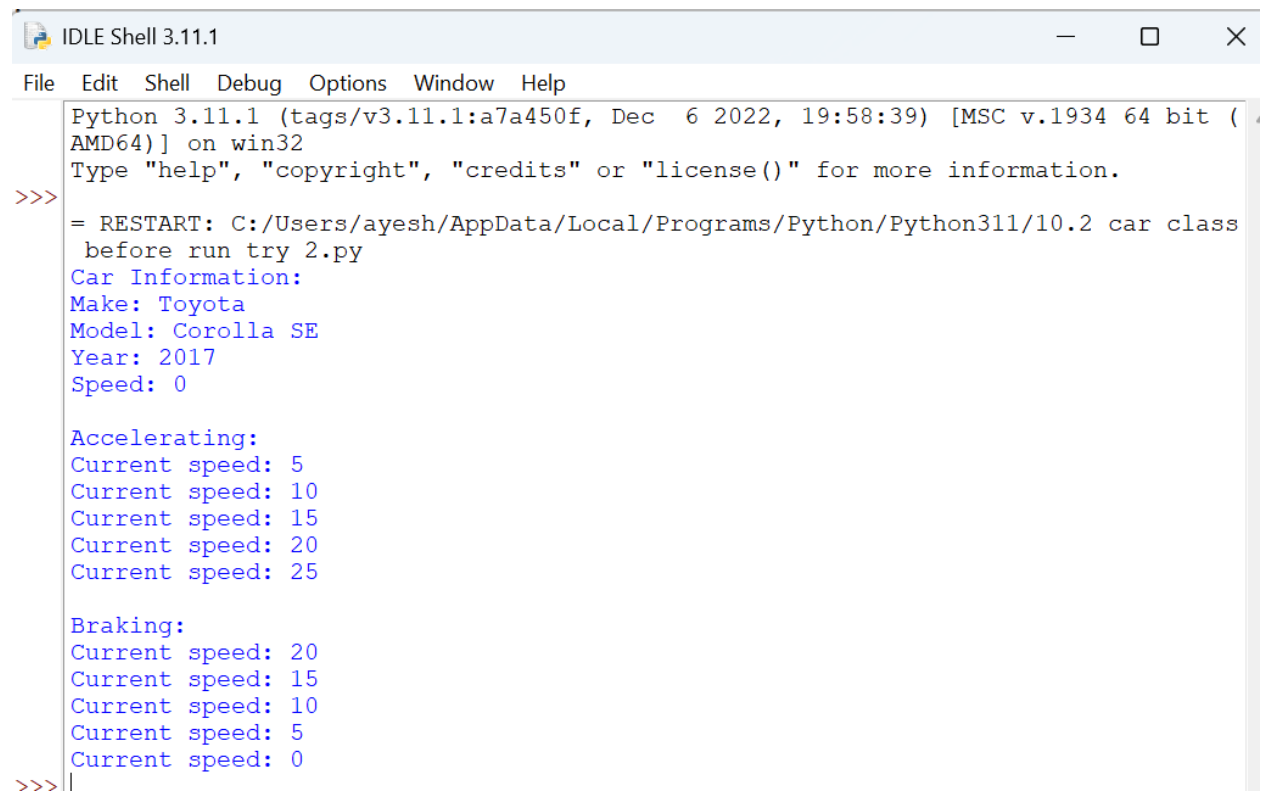
Current speed: 20

Current speed: 15

Current speed: 10

Current speed: 5

Current speed: 0



```
IDLE Shell 3.11.1
File Edit Shell Debug Options Window Help
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: C:/Users/ayesh/AppData/Local/Programs/Python/Python311/10.2 car class
before run try 2.py
Car Information:
Make: Toyota
Model: Corolla SE
Year: 2017
Speed: 0

Accelerating:
Current speed: 5
Current speed: 10
Current speed: 15
Current speed: 20
Current speed: 25

Braking:
Current speed: 20
Current speed: 15
Current speed: 10
Current speed: 5
Current speed: 0
>>> |
```