- Exercise 5-6 – Ice Cream Stand
    - Goal:
        - ❖ To practice class inheritance by extending an existing class (Restaurant) to create a specialized version (IceCreamStand).
    - Purpose:
        - ❖ The goal is to understand how a subclass can reuse and expand the behavior of a parent class while adding new attributes and methods specific to the subclass.
    - Algorithm / Steps:
        - ❖ Start from the Restaurant class defined in Exercise 5-4.
        - ❖ Create a new subclass IceCreamStand that inherits from Restaurant.
        - ❖ Add a new attribute flavors – a list of available ice-cream flavors.
        - ❖ Create a method display_flavors() that prints out all available flavors.
        - ❖ Instantiate IceCreamStand and call its methods to verify inheritance and new functionality.
    - Result:

ex5-6.py > ...

```python
class Restaurant():
    def __init__(self, rn, ct, number_served):
        self.flavors = []

    def describe_restaurant(self):
        print('Restaurant name: ', self.restaurant_name)
        print('Cuisine type: ', self.cuisine_type)

    def open_restaurant(self):
        print("The restaurant is open.")

    def set_number_served(self, n):
        self.number_served = n

    def increment_number_served(self, n):
        self.number_served += n

class IceCreamStand(Restaurant):
    def __init__(self, rn, ct="Ice Cream", number_served=0):
        super().__init__(rn, ct, number_served)
        self.flavors = ["vanilla", "chocolate", "strawberry", "mint"]

    def display_flavors(self):
        print("\nAvailable ice cream flavors:")
        for flavor in self.flavors:
            print("-", flavor)

ice_cream_stand = IceCreamStand("Happy Ice")
ice_cream_stand.describe_restaurant()
ice_cream_stand.display_flavors()
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 3

```
tathi@SGPLN-CG04200S9 MINGW64 /d/Master/python-for-engineer/05.Sun_Oct-05-2025_Morning
$ python ex5-6.py
Restaurant name:  Happy Ice
Cuisine type:  Ice Cream

Available ice cream flavors:
- vanilla
- chocolate
- strawberry
- mint
```

- Exercise 5-7 – Admin
    - Goal:

- ❖ To learn how to extend a user class to create an administrator class with additional permissions.

- Purpose:
  - ❖ This exercise demonstrates how subclasses can add new functionality in this case, defining admin privileges different from regular users.
- Algorithm / Steps:
  - ❖ Start with the User class from Exercise 5-5.
  - ❖ Create a subclass Admin that inherits from User.
  - ❖ Add a new attribute privileges – a list of permissions specific to administrators.
  - ❖ Implement a method show_privileges() to display these privileges.
  - ❖ Instantiate an Admin object and call the methods to confirm the privileges are shown correctly.
- Result:

```python
class User:

    def greet_user(self):
        print(f"Welcome {self.firstName} {self.lastName}")

    def increment_login_attempts(self):
        self.loginAttempts += 1
        print('loginAttempts: ', self.loginAttempts)

    def reset_login_attempts(self):
        self.loginAttempts = 0
        print('loginAttempts after reset: ', self.loginAttempts)

    def __str__(self):
        return f"{self.firstName} {self.lastName} (age: {self.age}, loginAttempts: {self.loginAttempts})"

class Admin(User):
    def __init__(self, firstName, lastName, age):
        super().__init__(firstName, lastName, age)
        self.privileges = ["can add post", "can delete post", "can ban user"]

    def show_privileges(self):
        print("\nAdmin privileges:")
        for p in self.privileges:
            print("-", p)

admin1 = Admin("Alice", "Smith", 30)
admin1.describe_user()
admin1.show_privileges()
```

```
PROBLEMS 3  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SPELL CHECKER 3

tathi@SGPLN-CG04200S9 MINGW64 /d/Master/python-for-engineer/05.Sun_Oct-05-2025_Morning
$ python ex5-7.py
Your information:
Alice
Smith
30

Admin privileges:
- can add post
- can delete post
- can ban user
```

- Exercise 5-8 – Privileges Class
    - Goal:
        - ❖ To understand and apply the concept of composition (a class containing another class as an attribute).
    - Purpose:
        - ❖ The goal is to refactor the previous exercise by separating the privileges attribute into its own class Privileges, improving clarity, modularity, and maintainability.
    - Algorithm / Steps:

- ❖ Create a new class Privileges that stores a list of admin privileges.
- ❖ Move the privilege-related code from Admin into this new class.
- ❖ Add a show_privileges() method to display the privileges.
- ❖ Modify AdminWithPrivileges to include a Privileges object as an attribute.
- ❖ Instantiate AdminWithPrivileges and call describe_user() and privileges.show_privileges().

- Result:

ex5-8.py > User > reset_login_attempts

```
class User:
        return f"{self.firstName} {self.lastName} (age: {self.age}, loginAttempts: {self.loginAttempts})"

class Admin(User):
    def __init__(self, firstName, lastName, age):
        super().__init__(firstName, lastName, age)
        self.privileges = ["can add post", "can delete post", "can ban user"]

    def show_privileges(self):
        print("\nAdmin privileges:")
        for p in self.privileges:
            print("-", p)

class Privileges:
    def __init__(self, privileges=None):
        if privileges is None:
            privileges = ["can add post", "can delete post", "can ban user"]
        self.privileges = privileges

    def show_privileges(self):
        print("\nPrivileges list:")
        for p in self.privileges:
            print("-", p)

class AdminWithPrivileges(User):
    def __init__(self, firstName, lastName, age):
        super().__init__(firstName, lastName, age)
        self.privileges = Privileges()


admin2 = AdminWithPrivileges("Bob", "Nguyen", "bob@example.com")
admin2.describe_user()
admin2.privileges.show_privileges()
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS SPELL CHECKER 3

```
tathi@SGPLN-CG04200S9 MINGW64 /d/Master/python-for-engineer/05.Sun_Oct-05-2025_Morning
$ python ex5-8.py
Your information:
Bob
Nguyen
bob@example.com

Privileges list:
- can add post
- can delete post
- can ban user
```