# TASK – WEEK 8

## TASK: Apply core object-oriented programming concepts to real-world scenarios

## 1. Shape area calculator with inheritance

- Define an abstract class ‘Shape’ with an abstract method ‘area()’.  
- Create two child classes: ‘Rectangle’ and ‘Circle’, each implementing its own version of ‘area()’.  
- Each class should be initialized with appropriate attributes (e.g. width/height or radius).  
- Add a docstring to each class describing what it represents and how it calculates the area.  
- In the main part of the program, print the docstring of each shape class to verify the documentation.  
- Create a list of shape objects and print their info using a loop.  
- Count how many rectangles and circles are in the list.

## 2. Bank account with encapsulation

- Create a class ‘BankAccount’ with a private balance attribute.  
- Use ‘@property’ and ‘@setter’ to allow reading and updating the balance, but prevent the balance from being set to a negative value.  
- Add a method ‘deposit(amount)’ and ‘withdraw(amount)’ that update balance safely.  
- Raise exceptions if invalid operations are attempted.  
- Create an object, test deposits, withdrawals, and invalid inputs.

## 3. Notification system with polymorphism

- Define two classes: ‘EmailNotification’ and ‘SMSNotification’.  
- Both should implement a method ‘send(message)’ that prints a different format of notification.  
- Write a function ‘send\_bulk(notifiers, message)’ that loops through any list of objects  
 and calls ‘.send()’ on them without checking their type.  
- Demonstrate that this works using duck typing.

## Note

- Tasks are not mandatory; they are voluntary.  
- Tasks are not time-limited when they need to be done – ideally, they should follow up on a weekly presentation, but not necessarily.  
- Saved .py files send to emails:  
 [bosko.nikolic@endava.com](mailto:bosko.nikolic@endava.com)  
 [djordje.munizaba@endava.com](mailto:djordje.munizaba@endava.com)