COMP8270
Programming for Artificial Intelligence

Class 5

Working with Python Classes

1. Create a class named `Account` with `account_number` and `balance` as instance attributes. Add a constructor that expects values for both attributes.

2. Add a method to the class `Account` named `deposit` that increases the account balance by a specified amount. The deposit should only operate with values greater than 0.

3. Add a method to the class `Account` named `withdraw` that decreases the account balance by a specified amount. The withdraw should only operate with values greater than 0 and smaller than the current balance of the account.

4. Modify your Account class so that both `account_number` and `balance` appear as "private" attributes. Define getter methods for each attribute.

5. Create a subclass of Account named `SavingsAccount` with an `interest` instance attribute. The initial value of interest should be specified in the constructor of the class – if no value is specified, it should use `0.02 (2%)` as a default value. Make sure that the constructor is calling the super class constructor.

6. Create a subclass of Account named `CurrentAccount` with an `overdraft` instance attribute. The initial value of overdraft should be specified in the constructor of the class – if no value is specified, it should use `£100` as a default value. Make sure that the constructor is calling the super class constructor.

7. Provide a specialised implementation (method overriding) for the `withdraw` method in the `CurrentAccount` class, which allows withdraws up to the limit of the overdraft – i.e., the account balance can be negative up to overdraft value.

8. Add a method to the class `SavingsAccount` named `annual_interest` that returns the estimated annual interest income of the account – e.g., if the balance of the account is `£100` and the interest is `2%`, the method would return `2`.

9. To prevent the creation of objects from the class `Account`, modify your implementation so that `Account` is an abstract class and add an abstract method name `description`, which should return a string representing the type of the account ("current" or "savings"). Implement the `description` method in each subclass.

10. Create a list of `Account` objects. The list should contain objects of both `SavingsAccount` and `CurrentAccount` classes. Then write a loop that identifies the number of accounts that are the absolute value of the overdraft is greater than the account balance.