**Unit 2**

Challenge 1Implement a class called BankAccount that represents a bank account. The class should have private attributes for account number, account holder name, and account balance. Include methods to deposit money, withdraw money, and display the account balance. Ensure that the account balance cannot be accessed directly from outside the class. Write a program to create an instance of the BankAccount class and test the deposit and withdrawal functionality.

class BankAccount:

    def \_\_init\_\_(self, account\_number, account\_holder, initial\_balance=0.0):

        self.\_account\_number = account\_number

        self.\_account\_holder = account\_holder

        self.\_account\_balance = initial\_balance

    def deposit(self, amount):

        self.\_account\_balance += amount

        return f"Deposited {amount} units. New balance: {self.\_account\_balance} units."

    def withdraw(self, amount):

        if amount <= self.\_account\_balance:

            self.\_account\_balance -= amount

            return f"Withdrew {amount} units. New balance: {self.\_account\_balance} units."

        else:

            return "Insufficient funds."

    def display\_balance(self):

        return f"Account balance for {self.\_account\_holder}: {self.\_account\_balance} units."

# Example usage

account = BankAccount("123456", "John Doe", 1000.0)

print(account.display\_balance())

print(account.deposit(500.0))

print(account.withdraw(200.0))

print(account.withdraw(1500.0))  # This should show "Insufficient funds."

Output:

Account balance for John Doe: 1000.0 units.

Deposited 500.0 units. New balance: 1500.0 units.

Withdrew 200.0 units. New balance: 1300.0 units.

Insufficient funds.

Challenge -2

- Implement a class called Player that represents a cricket player. The Player class should have a method called play() which prints "The player is playing cricket. Derive two classes, Batsman and Bowler, from the Player class. Override the play() method in each derived class to print "The batsman is batting" and "The bowler is bowling", respectively. Write a program to create objects of both the Batsman and Bowler classes and call the play() method for each object.

class Player:

    def play(self):

        print("The player is playing cricket.")

class Batsman(Player):

    def play(self):

        print("The batsman is batting.")

class Bowler(Player):

    def play(self):

        print("The bowler is bowling.")

# Creating objects and calling play() method

batsman = Batsman()

bowler = Bowler()

[batsman.play](http://batsman.play)()

[bowler.play](http://bowler.play)()

Output:

The batsman is batting.

The bowler is bowling.