```
# Define the base class Player
class Player:
    def play(self):
        print("The player is playing cricket.")

# Define the derived class Batsman
class Batsman(Player):
    def play(self):
        print("The batsman is batting.")

# Define the derived class Bowler
class Bowler(Player):
    def play(self):
        print("The bowler is bowling.")

# create objects of Batsman and Bowler classes
batsman = Batsman()
```

```
14 class Batsman(Player):
15     def play(self):
16         print("The batsman is batting.")
17
18 # Define the derived class Bowler
19 class Bowler(Player):
20     def play(self):
21         print("The bowler is bowling.")
22
23 # Create objects of Batsman and Bowler classes
24 batsman = Batsman()
25 bowler = Bowler()
26
27 # Call the play() method for each object
28 batsman.play()
29 bowler.play()
30
```

```
6
7
8  class BankAccount:
9
10     def __init__(self, account_number, account_holder_name, initial_balance=0.0):
11         self.__account_number = account_number
12         self.__account_holder_name = account_holder_name
13         self.__account_balance = initial_balance
14
15     def deposit(self, amount):
16         if amount > 0:
17             self.__account_balance += amount
18             print("Deposited ₹{}. New balance: ₹{}".format(amount
```

```
18        # self.__account_balance = self.__account_balance+amount
19        print("Deposited ₹{}. New balance: ₹{}".format(amount,
20                                                       self.__account_balance))
21      else:
22        print("Invalid deposit amount.")
23
24    def withdraw(self, amount):
25      if amount > 0 and amount <= self.__account_balance:
26        self.__account_balance -= amount
27        print("Withdrew ₹{}. New balance: ₹{}".format(amount,
28                                                      self.__account_balance))
29      else:
30        print("Invalid withdrawal amount or insufficient balance.")
31
32    def display_balance(self):
33      print("Account balance for {} (Account #{}): ₹{}".format(
34          self.__account_holder_name, self.__account_number,
35          self.__account_balance))
36
```

```
        self.__account_holder_name, self.__account_number,
        self.__account_balance))


# Create an instance of the BankAccount class
account = BankAccount(account_number="123456789",
                      account_holder_name="Hari Prabu",
                      initial_balance=5000.0)

# Test deposit and withdrawal functionality
account.display_balance()
account.deposit(500.0)
account.withdraw(200.0)
account.withdraw(20000.0)
account.display_balance()
```