1.Bank Management System

Program :

```
from abc import ABC, abstractmethod


# Abstraction
class BankAccount(ABC):


    def __init__(self, name, balance):
        self.name = name
        self.__balance = balance   # Encapsulation (private variable)


    # Getter method for balance
    def get_balance(self):
        return self.__balance


    # Deposit method
    def deposit(self, amount):
        self.__balance += amount


    # Abstract method (Polymorphism)
    @abstractmethod
    def calculate_interest(self):
        pass


# Inheritance
class SavingsAccount(BankAccount):
    def __init__(self, name, balance):
        super().__init__(name, balance)
        self.interest_rate = 0.04   # 4% interest

    # Polymorphism (method overriding)
    def calculate_interest(self):
        return self.get_balance() * self.interest_rate

# ---- Main Program ----

name = input("Enter Name: ")
balance = float(input("Enter Balance: "))

account = SavingsAccount(name, balance)

print("Balance:", account.get_balance())
print("Interest:", account.calculate_interest())
```

Output :

```
Enter Name: Naveen
Enter Balance: 15000
Balance: 15000.0
Interest: 600.0
```