***Here's a sample implementation of the assignment:***

**// BankAccount Class**

**class BankAccount {**

**constructor(owner, balance = 0) {**

**this.owner = owner;**

**this.balance = balance;**

**}**

**deposit(amount) {**

**this.balance += amount;**

**console.log(`${amount} deposited into ${this.owner}'s account.`);**

**}**

**withdraw(amount) {**

**if (amount <= this.balance) {**

**this.balance -= amount;**

**console.log(`${amount} withdrawn from ${this.owner}'s account.`);**

**} else {**

**console.log(`Insufficient funds for withdrawal from ${this.owner}'s account.`);**

**}**

**}**

**}**

**// SavingsAccount Class (Inheritance)**

**class SavingsAccount extends BankAccount {**

**constructor(owner, balance = 0, interestRate) {**

**super(owner, balance);**

**this.interestRate = interestRate;**

**}**

**addInterest() {**

**const interest = this.balance \* (this.interestRate / 100);**

**this.balance += interest;**

**console.log(`Interest added to ${this.owner}'s account: ${interest}`);**

**}**

**}**

**// CheckingAccount Class (Inheritance)**

**class CheckingAccount extends BankAccount {**

**constructor(owner, balance = 0, overdraft) {**

**super(owner, balance);**

**this.overdraft = overdraft;**

**}**

**withdraw(amount) {**

**if (amount <= this.balance + this.overdraft) {**

**this.balance -= amount;**

**console.log(`${amount} withdrawn from ${this.owner}'s account.`);**

**} else {**

**console.log(`Exceeded overdraft limit for ${this.owner}'s account.`);**

**}**

**}**

**}**

**// Testing**

**const savingsAcc = new SavingsAccount('Alice', 1000, 5);**

**savingsAcc.deposit(500);**

**savingsAcc.addInterest();**

**console.log(`Current balance for ${savingsAcc.owner}'s account: ${savingsAcc.balance}`);**

**const checkingAcc = new CheckingAccount('Bob', 500, 200);**

**checkingAcc.deposit(300);**

**checkingAcc.withdraw(800);**

**console.log(`Current balance for ${checkingAcc.owner}'s account: ${checkingAcc.balance}`);**

**// Testing inheritance**

**console.log(savingsAcc instanceof BankAccount); // true**

**console.log(checkingAcc instanceof BankAccount); // true**

***In this implementation:***

* **The BankAccount class represents a generic bank account with deposit and withdraw methods.**
* **The SavingsAccount class inherits from BankAccount and adds an addInterest method to calculate and add interest.**
* **The CheckingAccount class also inherits from BankAccount and overrides the withdraw method to handle overdraft protection.**
* **Instances of SavingsAccount and CheckingAccount are tested to confirm their inheritance from BankAccount using the instanceof operator.**

1. **BankAccount Class**

 This class represents a generic bank account.

 It has a constructor that initializes the owner's name (owner) and the account balance (balance).

 The deposit method adds the specified amount to the account balance.

 The withdraw method subtracts the specified amount from the account balance, checking for sufficient funds.

1. **SavingsAccount Class (Inheritance)**

 This class inherits from BankAccount, extending its functionality.

 It has a constructor that calls the parent class constructor (super) to initialize the owner's name, balance, and adds an interestRate property.

 The addInterest method calculates and adds interest to the account balance based on the annual interest rate.

1. **CheckingAccount Class (Inheritance)**

 Similar to SavingsAccount, this class also inherits from BankAccount.

 It adds an overdraft property to represent the overdraft limit.

 It overrides the withdraw method from BankAccount to handle overdraft protection. If the withdrawal amount exceeds the balance but is within the overdraft limit, it allows the withdrawal and sets the balance to negative.

1. **Testing**

 Instances of SavingsAccount and CheckingAccount are created, demonstrating their constructors and inheritance.

 Deposits and withdrawals are made to test the functionality of the deposit and withdraw methods.

 The addInterest method is called on savingsAcc to demonstrate adding interest.

 Finally, the instanceof operator is used to verify that instances of SavingsAccount and CheckingAccount are also instances of BankAccount, confirming inheritance.