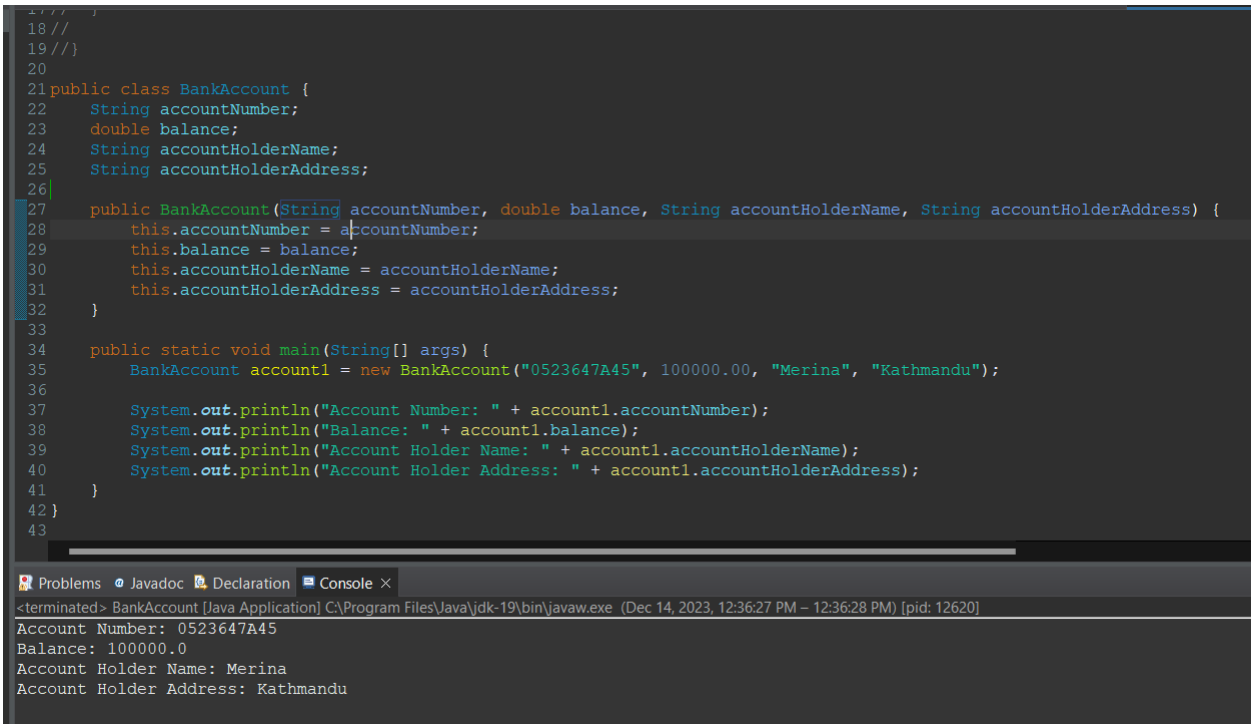
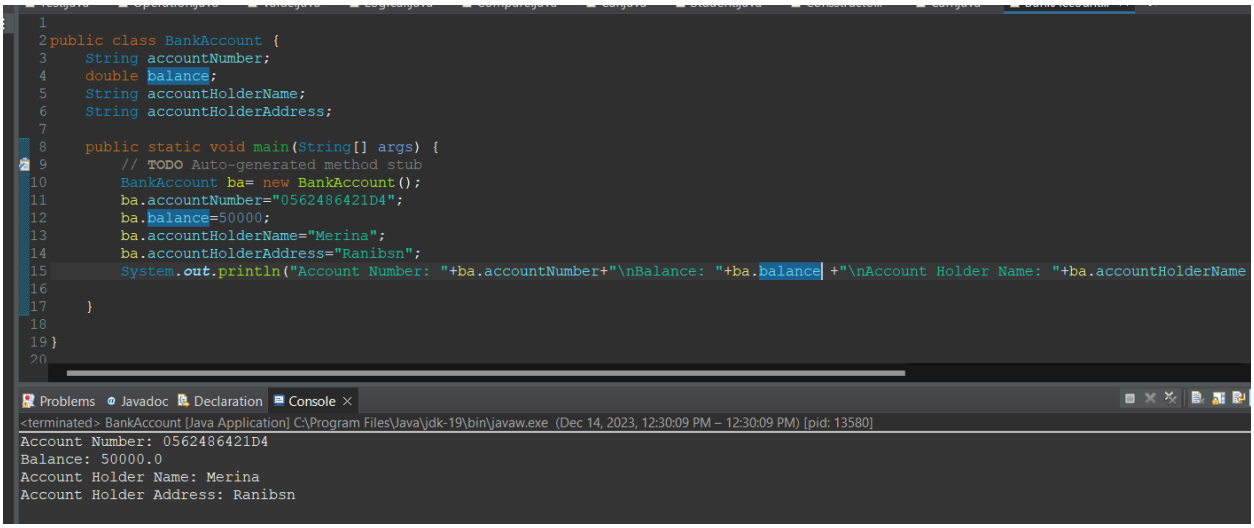
Basics of Classes and Objects:

1. Define a class named BankAccount with attributes accountNumber, balance, and accountHolderName, accountHolderAddress.



```
//
//}

public class BankAccount {
    String accountNumber;
    double balance;
    String accountHolderName;
    String accountHolderAddress;

    public BankAccount(String accountNumber, double balance, String accountHolderName, String accountHolderAddress) {
        this.accountNumber = accountNumber;
        this.balance = balance;
        this.accountHolderName = accountHolderName;
        this.accountHolderAddress = accountHolderAddress;
    }

    public static void main(String[] args) {
        BankAccount account1 = new BankAccount("0523647A45", 100000.00, "Merina", "Kathmandu");

        System.out.println("Account Number: " + account1.accountNumber);
        System.out.println("Balance: " + account1.balance);
        System.out.println("Account Holder Name: " + account1.accountHolderName);
        System.out.println("Account Holder Address: " + account1.accountHolderAddress);
    }
}

Problems  Javadoc  Declaration  Console ×
<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 12:36:27 PM – 12:36:28 PM) [pid: 12620]
Account Number: 0523647A45
Balance: 100000.0
Account Holder Name: Merina
Account Holder Address: Kathmandu
```

2. Create an object of this class and initialize its attributes.



```java
public class BankAccount {
    String accountNumber;
    double balance;
    String accountHolderName;
    String accountHolderAddress;

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        BankAccount ba= new BankAccount();
        ba.accountNumber="0562486421D4";
        ba.balance=50000;
        ba.accountHolderName="Merina";
        ba.accountHolderAddress="Ranibsn";
        System.out.println("Account Number: "+ba.accountNumber+"\nBalance: "+ba.balance +"\nAccount Holder Name: "+ba.accountHolderName

    }

}
```

Problems Javadoc Declaration Console

```
<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 12:30:09 PM – 12:30:09 PM) [pid: 13580]
Account Number: 0562486421D4
Balance: 50000.0
Account Holder Name: Merina
Account Holder Address: Ranibsn
```

Methods:

3. Create a method, **depositMoney()** in the BankAccount class to deposit money. Implement another method, **withdrawMoney()** to withdraw money. (The balance should also).

```java
public class BankAccount {
    int accountNumber;
    double balance;
    String accountHolderName;
    String accountHolderAddress;

    public BankAccount(int accountNumber, double balance, String accountHolderName, String accountHolderAddress) {
        this.accountNumber = accountNumber;
        this.balance = balance;
        this.accountHolderName = accountHolderName;
        this.accountHolderAddress = accountHolderAddress;
    }

    public void depositMoney(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid deposit amount. Please deposit an amount greater than 0.");
        }
    }

    public void withdrawMoney(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid withdrawal amount. Please check your balance and withdrawal amount.");
        }
    }
```

```java
    public static void main(String[] args) {
        // Create an instance of BankAccount1 and initialize its attributes
        BankAccount account = new BankAccount(254684564, 10000.00, "Reeya", "Riverdale");

        // Deposit and withdraw money
        account.depositMoney(5000.00);
        account.withdrawMoney(2000.00);
    }
}
```

Problems Javadoc Declaration Console

```
<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 12:47:01 PM – 12:47:01 PM) [pid: 17244
Deposited: $5000.0
New Balance: $15000.0
Withdrawn: $2000.0
New Balance: $13000.0
```

4. Create a class Lamp with attributes **isOn** to store boolean value. Also create a method **turnOn()** to turn on the light, and **turnOff()** to turn off the light and print the on status of the light.

```java
public class Lamp {
    private boolean isOn;

    public Lamp() {
        this.isOn = false;
    }

    public void turnOn() {
        isOn = true;
        System.out.println("Light is now ON.");
    }

    public void turnOff() {
        isOn = false;
        System.out.println("Light is now OFF.");
    }

    public boolean isOn() {
        return isOn;
    }

    public static void main(String[] args) {
        Lamp myLamp = new Lamp();
        System.out.println("Initial Status: " + (myLamp.isOn() ? "ON" : "OFF"));
        myLamp.turnOn();
        System.out.println("Status after turning on: " + (myLamp.isOn() ? "ON" : "OFF"));
        myLamp.turnOff();
        System.out.println("Status after turning off: " + (myLamp.isOn() ? "ON" : "OFF"));
    }
}
```

Problems Javadoc Declaration Console

<terminated> Lamp [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 1:07:37 PM – 1:07:37 PM) [pid: 14052]

```
Initial Status: OFF
Light is now ON.
Status after turning on: ON
Light is now OFF.
Status after turning off: OFF
```

Constructors:

5. Implement a parameterized constructor for the BankAccount class that initializes the account attributes. Create an object using this constructor.

```java
public class BankAccount {
    int accountNumber;
    double balance;
    String accountHolderName;
    String accountHolderAddress;

    public BankAccount(int accountNumber, double balance, String accountHolderName, String a
        this.accountNumber = accountNumber;
        this.balance = balance;
        this.accountHolderName = accountHolderName;
        this.accountHolderAddress = accountHolderAddress;
    }

    public void depositMoney(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid deposit amount. Please deposit an amount greater tha
        }
    }

    public void withdrawMoney(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid withdrawal amount. Please check your balance and wit
```

```java
    public void withdrawMoney(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid withdrawal amount. Please check your balance and wit
        }
    }

    public static void main(String[] args) {
        BankAccount account = new BankAccount(254684564, 10000.00, "Reeya", "Riverdale");

        // Deposit and withdraw money
        account.depositMoney(5000.00);
        account.withdrawMoney(2000.00);
    }
}
```

Problems Javadoc Declaration Console

<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 1:15:16 PM – 1:15:16 PM) [pid: 14324]

```
Deposited: $5000.0
New Balance: $15000.0
Withdrawn: $2000.0
New Balance: $13000.0
```

6. Implement a no-argument constructor that prints out "**User created!**" as soon as the instance of the user is created.

```
96
97public class BankAccount {
98    // Attributes
99    private String accountNumber;
100    private double balance;
101    private String accountHolderName;
102    private String accountHolderAddress;
103
104    // No-argument Constructor
105    public BankAccount() {
106        System.out.println("User created!");
107    }
108
109    // Parameterized Constructor
110    public BankAccount(String accountNumber, double balance, String accountHolderName, String accountHolderAddress) {
111        this.accountNumber = accountNumber;
112        this.balance = balance;
113        this.accountHolderName = accountHolderName;
114        this.accountHolderAddress = accountHolderAddress;
115    }
116
117    // Getter methods (unchanged)
118    public String getAccountNumber() {
119        return accountNumber;
120    }
121
122    public double getBalance() {
123        return balance;
124    }
```

```
126    public String getAccountHolderName() {
127        return accountHolderName;
128    }
129
130    public String getAccountHolderAddress() {
131        return accountHolderAddress;
132    }
133
134    // Setter methods (unchanged)
135    public void setBalance(double balance) {
136        this.balance = balance;
137    }
138
139    // Method to deposit money (unchanged)
140    public void depositMoney(double amount) {
141        if (amount > 0) {
142            balance += amount;
143            System.out.println("Deposited: $" + amount);
144            System.out.println("New Balance: $" + balance);
145        } else {
146            System.out.println("Invalid deposit amount. Amount should be greater than 0.");
147        }
148    }
149
150    // Method to withdraw money (unchanged)
151    public void withdrawMoney(double amount) {
152        if (amount > 0 && amount <= balance) {
153            balance -= amount;
154            System.out.println("Withdrawn: $" + amount);
155            System.out.println("New Balance: $" + balance);
156        } else {
```

```
            System.out.println("Invalid deposit amount. Amount should be greater than 0.");
        }
    }

    // Method to withdraw money (unchanged)
    public void withdrawMoney(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient funds.");
        }
    }

    public static void main(String[] args) {
        // Creating an object using the no-argument constructor
        BankAccount user = new BankAccount();

        // Creating another object using the parameterized constructor
        BankAccount account = new BankAccount("4579651354", 200000.00, "Alice Smith", "Kathmandu");

        // Displaying account information
        System.out.println("Account Number: " + account.getAccountNumber());
        System.out.println("Balance: $" + account.getBalance());
        System.out.println("Account Holder Name: " + account.getAccountHolderName());
        System.out.println("Account Holder Address: " + account.getAccountHolderAddress());
    }
}
```

Problems Javadoc Declaration Console

```
<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 1:20:37 PM – 1:20:37 PM) [pid: 18280]
User created!
Account Number: 4579651354
Balance: $200000.0
Account Holder Name: Alice Smith
Account Holder Address: Kathmandu
```

Constructor Overloading:

7. Create a class named ,"**Box**" with attributes **width**, **height**, and **depth.** Create multiple constructors for handling following object declarations:
    a. For a square, declare a constructor to take length only.
    b. For a rectangle, declare a constructor to take length, breadth, and height.

c. For no parameter, declare a no-argument constructor that sets **length = 10**, **breadth = 8**, and **height = 12**.

```java
public class Box {
    private double width;
    private double height;
    private double depth;

    public Box(double length) {
        this.width = length;
        this.height = length;
        this.depth = length;
    }

    public Box(double length, double breadth, double height) {
        this.width = length;
        this.height = breadth;
        this.depth = height;
    }

    public Box() {
        this.width = 10;
        this.height = 8;
        this.depth = 12;
    }

    public double getWidth() {
        return width;
    }

    public double getHeight() {
        return height;
    }
```

```
31
32    public double getDepth() {
33        return depth;
34    }
35
36    public static void main(String[] args) {
37        Box squareBox = new Box(5);
38        System.out.println("Square Box - Width: " + squareBox.getWidth() + ", Height: " + squareBox.getHeight() + ", Depth: " + squareBox
39        Box rectangularBox = new Box(10, 6, 8);
40        System.out.println("Rectangular Box - Width: " + rectangularBox.getWidth() + ", Height: " + rectangularBox.getHeight() + ", Depth
41        Box defaultBox = new Box();
42        System.out.println("Default Box - Width: " + defaultBox.getWidth() + ", Height: " + defaultBox.getHeight() + ", Depth: " + defaul
43    }
44}
45
```

Problems Javadoc Declaration Console

```
<terminated> Box [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 1:30:51 PM – 1:30:51 PM) [pid: 3744]
Square Box - Width: 5.0, Height: 5.0, Depth: 5.0
Rectangular Box - Width: 10.0, Height: 6.0, Depth: 8.0
Default Box - Width: 10.0, Height: 8.0, Depth: 12.0
```

Access Modifiers:

8. Set the balance attribute in the BankAccount class as private. Provide public getter and setter methods for the balance.

```
179//question 8
180public class BankAccount {
181    private double balance;
182
183    public BankAccount(int accountNumber, double balance, String accountHolderName, String accountHolderAddress) {
184        this.balance = balance;
185
186    }
187    public BankAccount() {
188        this.balance = 0.0;
189    }
190
191    public double getBalance() {
192        return balance;
193    }
194    public void setBalance(double balance) {
195        this.balance = balance;
196    }
197    public static void main(String[] args) {
198        BankAccount account1 = new BankAccount(123456789, 1000.00, "John Doe", "123 Main St, Cityville");
199        System.out.println("Current Balance: $" + account1.getBalance());
200        account1.setBalance(1500.00);
201        System.out.println("Updated Balance: $" + account1.getBalance());
202        }
203    }
204
205
```

Problems Javadoc Declaration Console

```
<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 1:38:47 PM – 1:38:47 PM) [pid: 10544]
Current Balance: $1000.0
Updated Balance: $1500.0
```

Encapsulation:

9. Create a class Address with private attributes street, city, and zipCode. Use encapsulation and provide getter and setter methods.

```java
public class Address {
    private String street;
    private String city;
    private String zipCode;

    public Address(String street, String city, String zipCode) {
        this.street = street;
        this.city = city;
        this.zipCode = zipCode;
    }


    public String getStreet() {
        return street;
    }

    public String getCity() {
        return city;
    }

    public String getZipCode() {
        return zipCode;
    }

    public void setStreet(String street) {
        this.street = street;
    }

    public void setCity(String city) {
        this.city = city;
    }
```

```java

    public void setStreet(String street) {
        this.street = street;
    }

    public void setCity(String city) {
        this.city = city;
    }

    public void setZipCode(String zipCode) {
        this.zipCode = zipCode;
    }

    public static void main(String[] args) {
        Address myAddress = new Address("123 Main Street", "Cityville", "12345");

        System.out.println("Street: " + myAddress.getStreet());
        System.out.println("City: " + myAddress.getCity());
        System.out.println("Zip Code: " + myAddress.getZipCode());

        myAddress.setStreet("456 Oak Avenue");
        myAddress.setCity("Townsville");
        myAddress.setZipCode("67890");

        System.out.println("\nUpdated Address:");
        System.out.println("Street: " + myAddress.getStreet());
        System.out.println("City: " + myAddress.getCity());
        System.out.println("Zip Code: " + myAddress.getZipCode());
    }
}

```

Problems Javadoc Declaration Console

```
<terminated> Address [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (Dec 14, 2023, 1:41:04 PM – 1:41:05 PM) [pid: 6180
Street: 123 Main Street
City: Cityville
Zip Code: 12345

Updated Address:
Street: 456 Oak Avenue
City: Townsville
Zip Code: 67890
```

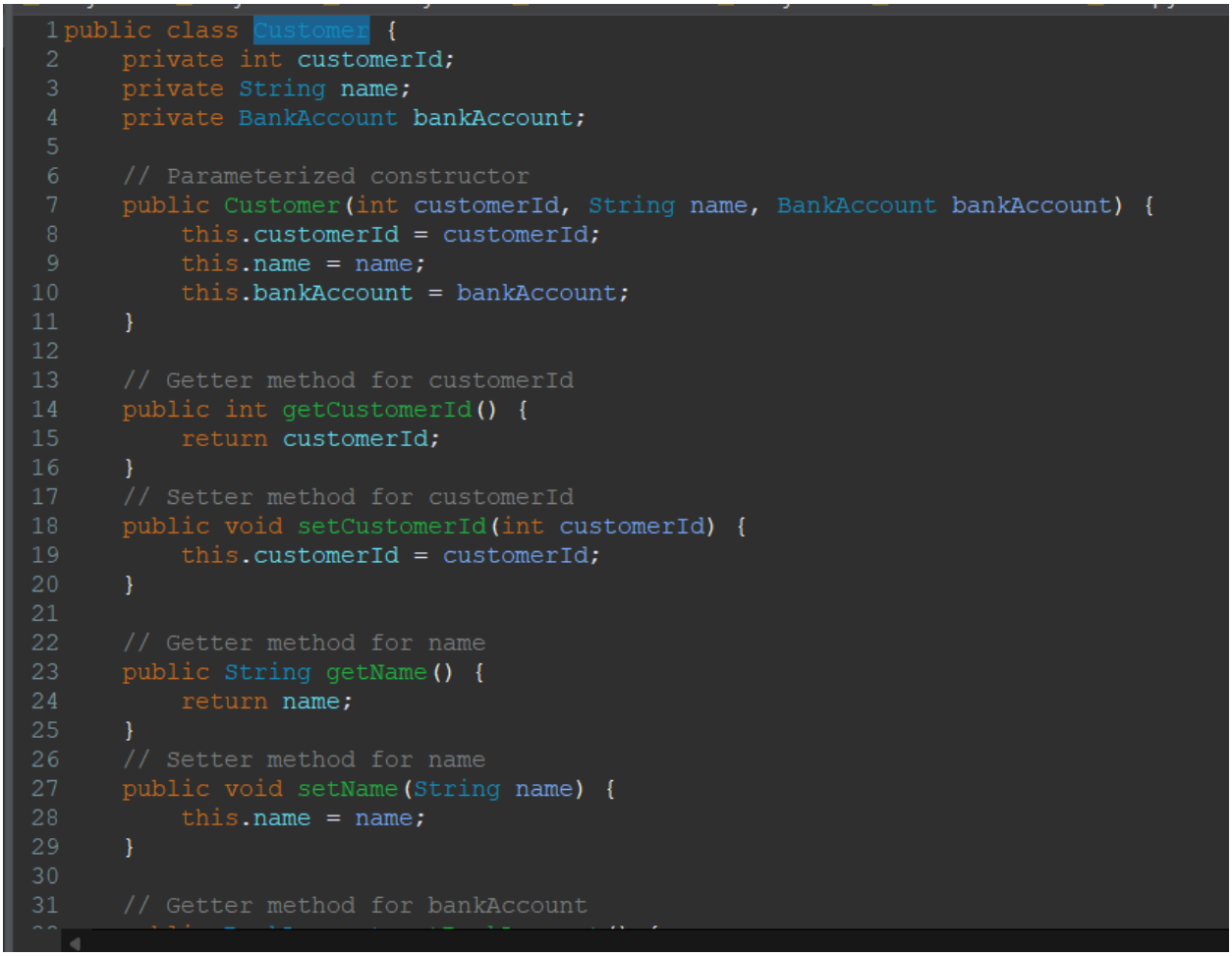
Combining Concepts:

10. Create a class Customer with private attributes customerId, name, and a BankAccount object. Implement a parameterized constructor and encapsulate the attributes. Provide getter and

setter methods. Instantiate multiple Customer objects with different values and demonstrate the use of getters and setters.

```
public class BankAccount {
    private double balance;

    public BankAccount( double balance) {
        this.balance = balance;
    }

    public BankAccount(int i, double d, String string, String string2) {
        // TODO Auto-generated constructor stub
    }

    public void depositMoney(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid deposit amount. Please deposit an amount greater than 0.");
        }
    }

    public void withdrawMoney(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid withdrawal amount. Please check your balance and withdrawal amount.");
        }
    }
```

```
    public double getBalance() {
        return balance;
    }
}
```



```java
public class Customer {
    private int customerId;
    private String name;
    private BankAccount bankAccount;

    // Parameterized constructor
    public Customer(int customerId, String name, BankAccount bankAccount) {
        this.customerId = customerId;
        this.name = name;
        this.bankAccount = bankAccount;
    }

    // Getter method for customerId
    public int getCustomerId() {
        return customerId;
    }
    // Setter method for customerId
    public void setCustomerId(int customerId) {
        this.customerId = customerId;
    }

    // Getter method for name
    public String getName() {
        return name;
    }
    // Setter method for name
    public void setName(String name) {
        this.name = name;
    }

    // Getter method for bankAccount
```

```java
    // Getter method for bankAccount
    public BankAccount getBankAccount() {
        return bankAccount;
    }

    // Setter method for bankAccount
    public void setBankAccount(BankAccount bankAccount) {
        this.bankAccount = bankAccount;
    }

    public static void main(String[] args) {
        // Create BankAccount objects for demonstration
        BankAccount account1 = new BankAccount(123456789, 1000.00, "John Doe", "123 Main St, Cityville");
        BankAccount account2 = new BankAccount(987654321, 2000.00, "Jane Doe", "456 Oak Ave, Townsville");

        // Create Customer objects with different values
        Customer customer1 = new Customer(1, "Alice", account1);
        Customer customer2 = new Customer(2, "Bob", account2);

        // Demonstrate the use of getters and setters
        System.out.println("Customer 1 ID: " + customer1.getCustomerId());
        System.out.println("Customer 1 Name: " + customer1.getName());
        System.out.println("Customer 1 Account Balance: $" + customer1.getBankAccount().getBalance());

        System.out.println();

        System.out.println("Customer 2 ID: " + customer2.getCustomerId());
        System.out.println("Customer 2 Name: " + customer2.getName());
        System.out.println("Customer 2 Account Balance: $" + customer2.getBankAccount().getBalance());

        // Modify values using setters
```

```java
        // Modify values using setters
        customer1.setCustomerId(101);
        customer1.setName("Alicia");
        customer1.getBankAccount().depositMoney(500.00);

        System.out.println();

        System.out.println("Modified Customer 1 ID: " + customer1.getCustomerId());
        System.out.println("Modified Customer 1 Name: " + customer1.getName());
        System.out.println("Modified Customer 1 Account Balance: $" + customer1.getBankAccount().getBalance());
    }
}
```

Problems Javadoc Declaration Console

```
<terminated> Customer [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe  (Dec 14, 2023, 1:56:33 PM – 1:56:33 PM) [pid: 14600]
Deposited: $500.0
New Balance: $500.0

Modified Customer 1 ID: 101
Modified Customer 1 Name: Alicia
Modified Customer 1 Account Balance: $500.0
```

Constructors and Overloading:

11. Implement multiple constructors for the BankAccount class with different parameter sets. Use constructor overloading to create objects with different initialization scenarios.

```java
public class BankAccount {
    // Attributes
    private String accountNumber;
    private double balance;
    private String accountHolderName;
    private String accountHolderAddress;

    // Constructor with full parameter set
    public BankAccount(String accountNumber, double balance, String accountHolderName, String accountHolderAddress) {
        this.accountNumber = accountNumber;
        this.balance = balance;
        this.accountHolderName = accountHolderName;
        this.accountHolderAddress = accountHolderAddress;
    }

    // Constructor with partial parameter set (ignoring address)
    public BankAccount(String accountNumber, double balance, String accountHolderName) {
        this(accountNumber, balance, accountHolderName, "N/A");
    }

    // Constructor with minimal parameter set (setting default values)
    public BankAccount(String accountNumber, double balance) {
        this(accountNumber, balance, "Unknown", "N/A");
    }

    // Getter methods (unchanged)
    public String getAccountNumber() {
        return accountNumber;
    }
```

```java

    public double getBalance() {
        return balance;
    }

    public String getAccountHolderName() {
        return accountHolderName;
    }

    public String getAccountHolderAddress() {
        return accountHolderAddress;
    }

    // Setter methods (unchanged)
    public void setBalance(double balance) {
        this.balance = balance;
    }

    // Method to deposit money (unchanged)
    public void depositMoney(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid deposit amount. Amount should be greater than 0.");
        }
    }

    // Method to withdraw money (unchanged)
    public void withdrawMoney(double amount) {
```

```java
    // Method to withdraw money (unchanged)
    public void withdrawMoney(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrawn: $" + amount);
            System.out.println("New Balance: $" + balance);
        } else {
            System.out.println("Invalid withdrawal amount or insufficient funds.");
        }
    }

    public static void main(String[] args) {
        // Creating objects using different constructors
        BankAccount fullParameters = new BankAccount("123456789", 1000.00, "John Doe", "123 Main Street");
        BankAccount partialParameters = new BankAccount("987654321", 2000.00, "Jane Smith");
        BankAccount minimalParameters = new BankAccount("456789012", 500.00);

        // Displaying account information
        System.out.println("Full Parameters:");
        System.out.println("Account Number: " + fullParameters.getAccountNumber());
        System.out.println("Balance: $" + fullParameters.getBalance());
        System.out.println("Account Holder Name: " + fullParameters.getAccountHolderName());
        System.out.println("Account Holder Address: " + fullParameters.getAccountHolderAddress());

        System.out.println("\nPartial Parameters:");
        System.out.println("Account Number: " + partialParameters.getAccountNumber());
        System.out.println("Balance: $" + partialParameters.getBalance());
        System.out.println("Account Holder Name: " + partialParameters.getAccountHolderName());
        System.out.println("Account Holder Address: " + partialParameters.getAccountHolderAddress());

        System.out.println("\nMinimal Parameters:");
```

```
327         System.out.println("\nPartial Parameters:");
328         System.out.println("Account Number: " + partialParameters.getAccountNumber());
329         System.out.println("Balance: $" + partialParameters.getBalance());
330         System.out.println("Account Holder Name: " + partialParameters.getAccountHolderName());
331         System.out.println("Account Holder Address: " + partialParameters.getAccountHolderAddress());
332 
333         System.out.println("\nMinimal Parameters:");
334         System.out.println("Account Number: " + minimalParameters.getAccountNumber());
335         System.out.println("Balance: $" + minimalParameters.getBalance());
336         System.out.println("Account Holder Name: " + minimalParameters.getAccountHolderName());
337         System.out.println("Account Holder Address: " + minimalParameters.getAccountHolderAddress());
338     }
339 }
340 
341 
342 
343 
```

Problems Javadoc Declaration Console ×

<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (Dec 14, 2023, 1:59:42 PM – 1:59:45 PM) [pid: 20316]

```
Minimal Parameters:
Account Number: 456789012
Balance: $500.0
Account Holder Name: Unknown
Account Holder Address: N/A
```