### Step 1: Create a New Android Project

1. Open Android Studio.

2. Click on New Project.

3. Choose Empty Activity and click Next.

4. Name the project SimpleBankingApp, set the package name, and select Java or Kotlin as the language.

5. Click Finish.

### Step 2: Define Classes and Objects (Encapsulation and Inheritance)

We will create a BankAccount class, which will demonstrate encapsulation and inheritance.

#### BankAccount.java

java

```
package com.example.simplebankingapp;


public class BankAccount {
   // Encapsulation: Private variables to prevent direct access
   private String accountHolder;
   private double balance;


   // Constructor (Object creation)
   public BankAccount(String accountHolder, double initialBalance) {
      this.accountHolder = accountHolder;
      this.balance = initialBalance;
   }


   // Getter method to check balance
   public double getBalance() {
      return balance;
   }
```

```
    // Method to deposit amount

    public void deposit(double amount) {

        if (amount > 0) {

            balance += amount;

        }

    }


    // Method to withdraw amount

    public boolean withdraw(double amount) {

        if (amount > 0 && amount <= balance) {

            balance -= amount;

            return true;

        } else {

            return false; // Insufficient balance

        }

    }

}
```

#### SavingsAccount.java (Inheritance)

We will create a SavingsAccount class that inherits from the BankAccount class to demonstrate inheritance.

java

```
package com.example.simplebankingapp;


// Inheritance: SavingsAccount inherits from BankAccount

public class SavingsAccount extends BankAccount {


    private double interestRate;
    // Constructor with inheritance
    public SavingsAccount(String accountHolder, double initialBalance, double interestRate) {
        super(accountHolder, initialBalance); // Call to parent class constructor
        this.interestRate = interestRate;
    }

    // Method to apply interest to the account balance
    public void applyInterest() {
        double interest = getBalance()  interestRate / 100;
        deposit(interest);
    }
}
```

### Step 3: Create the Main Activity Layout (XML)

Create the user interface that will allow users to interact with the app.

#### activity_main.xml

xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/tv_balance"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Balance: $0.00"
        android:textSize="18sp"
        android:textColor="@android:color/black"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:padding="16dp"/>

    <EditText
        android:id="@+id/et_amount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Enter amount"
        android:inputType="numberDecimal"
        app:layout_constraintTop_toBottomOf="@id/tv_balance"
        app:layout_constraintStart_toStartOf="parent"
        android:padding="16dp"/>

    <Button
        android:id="@+id/btn_deposit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Deposit"
        app:layout_constraintTop_toBottomOf="@id/et_amount"
        app:layout_constraintStart_toStartOf="parent"
        android:padding="16dp"/>

    <Button
        android:id="@+id/btn_withdraw"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:text="Withdraw"
        app:layout_constraintTop_toBottomOf="@id/btn_deposit"
        app:layout_constraintStart_toStartOf="parent"
        android:padding="16dp"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

### Step 4: Implement Main Activity Logic
Now, link the buttons and user inputs to interact with the BankAccount object.

#### MainActivity.java
java
```
package com.example.simplebankingapp;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

    private BankAccount bankAccount;
    private TextView tvBalance;
    private EditText etAmount;
    private Button btnDeposit, btnWithdraw;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize the BankAccount object
        bankAccount = new BankAccount("John Doe", 1000.0);  // Initial balance of $1000

        // Bind UI elements to variables
        tvBalance = findViewById(R.id.tv_balance);
        etAmount = findViewById(R.id.et_amount);
        btnDeposit = findViewById(R.id.btn_deposit);
        btnWithdraw = findViewById(R.id.btn_withdraw);

        // Update balance display
        updateBalanceDisplay();

        // Deposit action
        btnDeposit.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                double amount = getAmountFromInput();
                if (amount > 0) {
                    bankAccount.deposit(amount);
                    updateBalanceDisplay();
                    Toast.makeText(MainActivity.this, "Deposited $" + amount,
Toast.LENGTH_SHORT).show();
                }
            }
        });

        // Withdraw action
```

```java
        btnWithdraw.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                double amount = getAmountFromInput();
                if (amount > 0 && bankAccount.withdraw(amount)) {
                    updateBalanceDisplay();
                    Toast.makeText(MainActivity.this, "Withdrew $" + amount, Toast.LENGTH_SHORT).show();
                } else {
                    Toast.makeText(MainActivity.this, "Insufficient balance", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }

    // Helper function to get amount from EditText
    private double getAmountFromInput() {
        String amountStr = etAmount.getText().toString();
        if (!amountStr.isEmpty()) {
            return Double.parseDouble(amountStr);
        } else {
            Toast.makeText(this, "Enter a valid amount", Toast.LENGTH_SHORT).show();
            return 0;
        }
    }

    // Update balance display
    private void updateBalanceDisplay() {
        tvBalance.setText("Balance: $" + bankAccount.getBalance());
    }
}
```

### Step 5: Run the Application

- Connect your Android device or start an emulator in Android Studio.

- Click Run to build and deploy the application.

### Concepts Demonstrated:

- Classes and Objects: BankAccount and SavingsAccount demonstrate object creation.

- Encapsulation: Balance is encapsulated in the BankAccount class with public methods to modify it.

- Inheritance: SavingsAccount inherits from BankAccount, showing code reuse.