The following is an example of a ***bankAccount*** class using encapsulation to include all methods (including the main method) to create a program to manage a bank account:

```
public class bankAccount  {

    /** 3 Instance variables
      *   1 double variable balance, 2 String variables pw and username.      */

    public bankAccount(String a)  {

        /** Constructor Method
          * Default state variable pw set to XYZ, Default state variable balance set to zero
          * State variable username set to user-defined name that is sent as a parameter          */
    }

    public void editPW( )  {

        /** Mutator Method
          * Ask and grab user input for a new password.  When done, ask the user to re-enter the new password.
          * Pass both user entries to the verifyPW( ) method to make sure that they match.
          * If they match, change the account password and tell the user that the password was changed.
          * If they do not match, do not change the password.  Tell the user there was no match and no change  */
    }

     public boolean verifyPW(String a, String b){

        /** Accessor Method
          * Use the user's new password and re-entry as parameters to check to see if they match.
          * If they match return true.  If they do not match, return false.                          */
    }

    public void deposit( )  {

        /** Mutator Method
          * Ask the user for the deposit amount and add the deposit amount to the account balance      */
    }

    public void withdraw( )  {

        /** Mutator Method
          * Ask the user for the withdraw amount and subtract the amount from the account balance      */
    }

    public double getBalance( )  {

        /** Accessor Method - return the current account balance      */
    }

    public String getPW( )  {

        /** Accessor Method – return the current account password      */
    }

    public String getUsername(){

        /** Accessor Method – return the current username      */
    }

    public void changeUsername( )  {

        /** Mutator Method
          * Ask the user for a new username and change the account username      */
    }
```

/* Continued on Next Page */

```
/** The main method is shown below. Read the following code to help you better
  * understand how to complete the previous methods. Create a class called bankAccount
  * and complete this entire process to make a program to manage a bank account.  */

public static void main(String[] args) {
        bankAccount myAccount = new bankAccount("MRH");
        int choice=0;
        while(choice!=8) {
            System.out.println("Welcome to your bank. Which would you like to do:");
            System.out.println("1. Edit Password");
            System.out.println("2. Deposit Money");
            System.out.println("3. Withdraw Money");
            System.out.println("4. Get Current Balance");
            System.out.println("5. Retrieve Password");
            System.out.println("6. Get Username");
            System.out.println("7. Change Username");
            System.out.println("8. Exit Account");
            System.out.print("Enter your selection: ");

            Scanner getChoice = new Scanner(System.in);
            choice=getChoice.nextInt( );

            if(choice==1)
                myAccount.editPW( );
            else if(choice==2)
               myAccount.deposit( );
            else if(choice==3)
                myAccount.withdraw( );
            else if(choice==4)
                System.out.println("\n\nCurrent balance: "+myAccount.getBalance());
             else if(choice==5)
                System.out.println("\n\nCurrent password: "+myAccount.getPW());
            else if(choice==6)
                System.out.println("\n\nUsername: "+myAccount.getUsername());
            else if(choice==7)
                myAccount.changeUsername( );
            else if(choice==8) {
                System.out.println("Program Exited!\n\n");
                break;
            }
            else
                System.out.println("Not a valid selection!");
        }
    }
}
```

Directions: Use the above code and structure to help you develop the ***bankAccount*** class. Complete the methods so that this program becomes operational and successfully completes all of the tasks that it is intended to do. Be sure to test your ***bankAccount*** class and to completely understand how it functions.