In this file, I review the code and changes the variable name, function name, class name that which are not following proper Naming Conventions.

The Corrections are added as Note near the incorrect names.

```
class LoginActivity : Activity() {
   lateinit var input : EditText
   lateinit  var  password : EditText
   lateinit var dbHelper : DBHelper
   var pswdEnc = PasswordEncryption()
/* NOTE :
For variable,
Bad name : pswdEnc, Good name : passwordEncrypter

For Class Name,
Bad name : PasswordEncryption , Good name : PasswordEncryptor
*/
   companion object {
      var sharedPreFile : String = "ShareWithPreference"
 /* NOTE  :
For variable name,
Bad name : sharedPreFile, Good name : sharedPreferenceFile
*/
   }

   override fun onCreate(savedInstanceState: Bundle?) {

      super.onCreate(savedInstanceState)
     setContentView(R.layout.activity_login)

      dbHelper = DBHelper(this,null)
      val signup_link = findViewById<TextView>(R.id.signup_link) as TextView
      val login_button = findViewById<Button>(R.id.login_button) as Button
      /*NOTE :
         The above variable are using snake_case for naming, generally we use
         camelCase for naming the variable
```

```
        Bad name : signup_link, login_button
        Good name : signupLink, loginButton
      */

      signup_link?.setOnClickListener{
         val intent = Intent(this, SignupActivity::class.java)
         startActivity(intent)

      }
      login_button?.setOnClickListener{
         getLoginProcess()
      }
   }


/*NOTE :
For function name,
Bad name : getLoginProcess, Good name : processLoginDetails
*/
fun getLoginProcess()
   {
      input = findViewById<EditText>(R.id.input_user) as EditText
      password = findViewById<EditText>(R.id.input_password) as EditText

      if(input.text.toString()!= "" && password.text.toString() != "")
      {
         if(isEmailValid(input.text.toString()))
         {
            val cursor : Cursor =
dbHelper.getUserByEmail(input.text.toString(),pswdEnc.encodeData(password.text.toSt
ring()))
            try {
               if (cursor!!.moveToFirst()) {

                  if (cursor == null || cursor.count == -1) {
                     Log.e("Bye", "User not found by email")
                     Toast.makeText(this,"Check Username and
```

```
password",Toast.LENGTH_SHORT).show()
                    input.text.clear()
                    password.text.clear()
                } else {
                    val sharedPref: SharedPreferences =
this.getSharedPreferences(sharedPreFile, Context.MODE_PRIVATE)
                    val sharedEmail =
cursor.getString(cursor.getColumnIndex(DBHelper.COLUMN_EMAILID))
                    val sharedMobile =
cursor.getString(cursor.getColumnIndex(DBHelper.COLUMN_MOBILE))
                    val sharedName =
cursor.getString(cursor.getColumnIndex(DBHelper.COLUMN_NAME))
/*NOTE :
For variables,
Bad names : sharedPref, sharedEmail, sharedMobile, sharedName

Good names : sharedPreference, currentUserEmailId, currentUserMobile,
currentUserName

since I used shared preference to store currently logged in user.
*/
                    val editor: SharedPreferences.Editor = sharedPref.edit()
                    editor.putString("EMAIL_KEY", sharedEmail)
                    editor.putString("MOBILE_KEY", sharedMobile)
                    editor.putString("NAME_KEY", sharedName)
                    editor.apply()
                    editor.commit()
                    val intent = Intent(this, TabbedMainActivity::class.java)
                    startActivity(intent)
                    finish()

                }
            }
            }
            catch(e : Exception) {
                Log.e("exception in login  ", "  in catch $e")
            }
```

```
        }

        else if(isMobileNumber(input.text.toString()))
        {
            val cursor : Cursor =
dbHelper.getUserByMobile(input.text.toString(),pswdEnc.encodeData(password.text.toS
tring()))
            try{
                if(cursor!!.moveToFirst())
                {
                    if(cursor == null || cursor.count == -1)
                    {
                        Log.e("Bye","User not found by mobile")
                        //Toast.makeText(this,"")
                    }
                    else{
                        val sharedPref : SharedPreferences =
this.getSharedPreferences(sharedPreFile, Context.MODE_PRIVATE)
                        val sharedEmail =
cursor.getString(cursor.getColumnIndex(DBHelper.COLUMN_EMAILID))
                        val sharedMobile =
cursor.getString(cursor.getColumnIndex(DBHelper.COLUMN_MOBILE))
                        val sharedName =
cursor.getString(cursor.getColumnIndex(DBHelper.COLUMN_NAME))
/*NOTE :
For variables,
Bad names : sharedPref, sharedEmail, sharedMobile, sharedName

Good names : sharedPreference, currentUserEmailId, currentUserMobile,
currentUserName
*/
                        val editor:SharedPreferences.Editor =  sharedPref.edit()
                        editor.putString("EMAIL_KEY",sharedEmail)
                        editor.putString("MOBILE_KEY",sharedMobile)
                        editor.putString("NAME_KEY",sharedName)
                        editor.apply()
                        editor.commit()
```

```
                    val intent = Intent(this, TabbedMainActivity::class.java)
                    startActivity(intent)
                    finish()
                }
            }
        }
        catch(e : Exception){
            Log.e("exception ", " $e")
        }
    }
  }


/*NOTE :
For function name,
Bad name : isEmailValid , Good name : isValidEmailId
*/
   fun isEmailValid(email: String): Boolean {
      return android.util.Patterns.EMAIL_ADDRESS.matcher(email).matches()
   }

   fun isMobileNumber(mobile : String) : Boolean{
      return android.util.Patterns.PHONE.matcher(mobile).matches()
   }

}
```

References :

https://kotlinlang.org/docs/coding-conventions.html#naming-rules

https://www.oracle.com/java/technologies/javase/codeconventions-namingconventions.html

https://www.freecodecamp.org/news/clean-coding-for-beginners/

https://dzone.com/articles/naming-conventions-from-uncle-bobs-clean-code-phil

https://medium.com/@pabashani.herath/clean-code-naming-conventions-4cac223de3c6

https://thecoderoad.blog/2020/03/29/clean-code-naming-conventions/