**Bad Practice:**

**private fun** getContactFromInputFields() : ContactDataClass?{  
 **val** firstName:String=findViewById<EditText>(R.id.*first\_name*).*text*.toString().*trim*()  
 **val** lasName:String=findViewById<EditText>(R.id.*last\_name*).*text*.toString().*trim*()  
 **val** number=ArrayList<String>()  
 **val** number1:String=findViewById<EditText>(R.id.*number1*).*text*.toString().*trim*()

// code duplication

**if** (number1.*isNotEmpty*()) {  
 **if** (!validateMobileNumber(number1)) {  
 Toast.makeText(**this**, *resources*.getString(R.string.*invalid\_number1*), Toast.*LENGTH\_SHORT*).show()  
 **return null**  
}  
 }  
 **val** number2=findViewById<EditText>(R.id.*number2*).*text*.toString().*trim*()  
 **if** (number2.*isNotEmpty*()) {  
 **if** (!validateMobileNumber(number2)) {  
 Toast.makeText(**this**, *resources*.getString(R.string.*invalid\_number2*), Toast.*LENGTH\_SHORT*).show()  
 **return null**  
}  
 }

//

**val** email:String=findViewById<EditText>(R.id.*email*).*text*.toString().*trim*()  
 **if** (email.*isNotEmpty*()) {  
 **if** (!validateEmail(email)) {  
 Toast.makeText(**this**, *resources*.getString(R.string.*invalid\_email*), Toast.*LENGTH\_SHORT*).show()  
 **return null**  
}  
 }

// dead code  
 **val** bitmap=(findViewById<ImageView>(R.id.*user\_ic*).*drawable* **as** BitmapDrawable).*bitmap*  
**val** stream=ByteArrayOutputStream() bitmap.compress(Bitmap.CompressFormat.*JPEG*,90,stream)

//

**if**(number1==**""**) {  
 number.add(*resources*.getString(R.string.*empty*))  
 } **else** {  
 number.add(number1)  
 }  
 **if**(number2==**""**) {  
 number.add(*resources*.getString(R.string.*empty*))  
 } **else** {  
 number.add(number2)  
 }  
 **return** ContactDataClass(firstName = firstName, lastName = lasName,number=number,email=email, profileImage = firstName+lasName+*resources*.getString(  
 R.string.*image\_format*  
))  
}

Good Practice:

// SRP

// Functions should be small ( 20 or 30 lines max)

// Don't use long ***if, else, switch*** and ***while*** statements

**private fun** isValidMobileNumber(mobileNumber:String, messageId:Int): Boolean{  
 **val** number1:String=findViewById<EditText>(R.id.*number1*).*text*.toString().*trim*()  
 **if** (number1.*isNotEmpty*()) {  
 **if** (!validateMobileNumber(number1)) {  
 Toast.makeText(**this**, *resources*.getString(messageId), Toast.*LENGTH\_SHORT*).show()  
 **return false**  
} **else** {  
 **return true**  
}  
 }  
 **return false**  
}

// SRP  
 **private fun** getMobileNumbers(): ArrayList<String>{  
 **val** number1 = findViewById<EditText>(R.id.*number1*).*text*.toString().*trim*()  
 **val** number2 = findViewById<EditText>(R.id.*number2*).*text*.toString().*trim*()  
 **val** number=ArrayList<String>()  
 **if**(number1 == **""**) {  
 number.add(*resources*.getString(R.string.*empty*))  
 } **else** {  
 number.add(number1)  
 }  
 **if**(number2 == **""**) {  
 number.add(*resources*.getString(R.string.*empty*))  
 } **else** {  
 number.add(number2)  
 }  
 **return** number  
 }  
  
 **private fun** getContactFromInputFields(): ContactDataClass?{  
 **val** firstName:String=findViewById<EditText>(R.id.*first\_name*).*text*.toString().*trim*()  
 **val** lasName:String=findViewById<EditText>(R.id.*last\_name*).*text*.toString().*trim*()  
 **val** number1 = findViewById<EditText>(R.id.*number1*).*text*.toString().*trim*()  
 **val** number2 = findViewById<EditText>(R.id.*number2*).*text*.toString().*trim*()  
 **if** (!isValidMobileNumber(number1, R.string.*invalid\_number1*)) {  
 **return null**  
}  
 **if**(!isValidMobileNumber(number2, R.string.*invalid\_number2*)) {  
 **return null**  
}  
 **val** email:String=findViewById<EditText>(R.id.*email*).*text*.toString().*trim*()  
 **if** (email.*isNotEmpty*()) {  
 **if** (!validateEmail(email)) {  
 Toast.makeText(**this**, *resources*.getString(R.string.*invalid\_email*), Toast.*LENGTH\_SHORT*).show()  
 **return null**  
}  
 }  
**val** number = getMobileNumbers()  
 **return** ContactDataClass(firstName = firstName, lastName = lasName,number=number,email=email, profileImage = firstName+lasName+*resources*.getString(  
 R.string.*image\_format*  
))  
 }