Functions

1. **Encapsulate Conditionals in Functions**

   Bad practice:

```
buttonSave.setOnClickListener {
  if (itemName.text.isNotEmpty() && radiobuttonCategory.checkedRadioButtonId != -1 &&
((checkboxRegular.isChecked && textviewRegularPrice.text.isNotEmpty()) ||
(checkboxMedium.isChecked && textviewMediumPrice.text.isNotEmpty()) ||
(checkboxLarge.isChecked && textviewLargePrice.text.isNotEmpty())))
      saveItem()
  else
      Toast.makeText(
         context,
         getString(R.string.save_error_message),
         Toast.LENGTH_SHORT
      ).show()
}
```

   Good practice:

```
buttonSave.setOnClickListener {  it: View!
    if (validate())
          saveItem()
    else
        Toast.makeText(
           context,
           "Please enter all the required fields",
           Toast.LENGTH_SHORT
        ).show()
}
```

```
fun validate(): Boolean{
    if(itemName.text.isEmpty())
        return false
    else if(radiobuttonCategory.checkedRadioButtonId == -1)
          return false
    else if((checkboxRegular.isChecked && textviewRegularPrice.text.isEmpty())
          return false
    return true
}
```

2. **Do not Repeat**

   Bad practice:

   Repeating the same code in more than one place.

```
if (orders.isNotEmpty()) {
    with(binding) { this: FragmentOrdersBinding
        ordersHistoryView.visibility = View.VISIBLE
        noOrdersView.visibility = View.GONE
    }
```

```
val orders = databaseHelper.filterOrdersByDate(selectedDate)
with(binding) { this: FragmentOrdersBinding
    ordersHistoryView.visibility = View.VISIBLE
    noOrdersView.visibility = View.GONE
}
recyclerView.adapter = OrdersAdapter(orders, ADMIN,requireContext())
```

Good Practice:

```
if (orders.isNotEmpty()) {
    setOrdersHistoryScreen()
    recyclerView.adapter = OrdersAdapter(orders, USER,requireContext())
} else {
    setEmptyOrdersScreen()
}
```

```
val orders = databaseHelper.filterOrdersByDate(selectedDate)
setOrdersHistoryScreen()
recyclerView.adapter = OrdersAdapter(orders, ADMIN,requireContext())
```

```
private fun setOrdersHistoryScreen() {
    with(binding) { this: FragmentOrdersBinding
        ordersHistoryView.visibility = View.VISIBLE
        noOrdersView.visibility = View.GONE
    }

}
```

3. A Function should have only one responsibility.
   Bad practice:

```
public void displayUser(int userId){
    Users user=null;
    for (Users u:LibraryDb.getUsers()) {
        if(userId==u.getUserId())
            user=u;
    }
    System.out.println("User Id: " + user.getUserId() + ", User Name: " + user.getName() +
}
```

displayUser function should be responsible only for displaying the user but here it is getting user from DB and Displaying.

Good Practice:

```
public Users getCurrentUser(int userId) {
    Users currentUser=null;
    for (Users u:LibraryDb.getUsers()) {
        if(userId==u.getUserId())
            currentUser=u;
    }

    return currentUser;
}

public void displayUser(int userId){
    Users user = getCurrentUser(userId);
    System.out.println("User Id: " + user.getUserId() + ", User Name: " + user.getName() +
}
```

4. **Do not use Flag Arguments**

   Bad Practice:

```
private void setArgumentValues(Bundle args)
{
 isTaskEdit = args.getInt(KEY_TASK_TYPE,TASK_ADD_EVENT) == TASK_EDIT_EVENT;
 if(isTaskEdit)
 {
   editZuId = args.getString(CalendarUtil.KEY_EVENT_HOST_ZUID);
   mEventTitleString = args.getString("title");//No I18N
   mIsAllDayEvent = args.getBoolean("isAll");//No I18N
   mStartTimeMillis = args.getLong(CalendarUtil.KEY_START_TIME_MILLIS);
   mEndTimeMillis = args.getLong(CalendarUtil.KEY_END_TIME_MILLIS);
   mLocationString = args.getString("location");//NO I18N
   mDescriptionString = args.getString("desc");//No I18N
   mAType = args.getInt("aType",-1);//NO I18N
   mRType = args.getInt("rType",-1);//NO I18N
   mAlarmTimeString = args.getString("alarmTime");//NO I18N
   mRepeatString = args.getString("repeat");//NO I18N
   mOriginalAttendeesString = args.getString("attendees"); //No I18N
   mEventKey = args.getString("eKey");//NO I18N
   mEventId = args.getString(ZMailConstants.KEY_EVENT_ID);
   mCurrentCalendarId = args.getString("calId");
   mCurrentCalendarName = args.getString("calName");//No I18N
 }
 else
 {
   mStartTimeMillis = args.getLong(CalendarUtil.KEY_SELECTED_DATE_MILLIS);
```

```
    mEventTitleString = "";
    mLocationString = mAlarmTimeString = mRepeatString = mOriginalAttendeesString ="";
    mIsAllDayEvent = false;
    mEventKey = mEventId = "";

    Calendar calendar = Calendar.getInstance();
    calendar.setTimeInMillis(mStartTimeMillis);
    calendar.add(Calendar.HOUR_OF_DAY,1);
    calendar.set(Calendar.MINUTE,0);
    calendar.set(Calendar.SECOND,0);
    mStartTimeMillis = calendar.getTimeInMillis();
    calendar.add(Calendar.HOUR_OF_DAY,1);
    mEndTimeMillis = calendar.getTimeInMillis();
  }
}
```

Good Practice:

Create two functions setValuesForEditTask() and setValuesForAddTask().

```
isTaskEdit = args.getInt(KEY_TASK_TYPE,TASK_ADD_EVENT) == TASK_EDIT_EVENT;

if(isTaskEdit)
     setValuesForEditTask()
  else
     setValuesForAddTask()
```