Assignment1

40 points:

- Using vertical LinearLayout for overall layout of main activity. Because each view is set up-to-down, so LinearLayout will be easy way to arrange them.
- Using a TextView to set the general title: To do list. And set gravity central horizontal, bold for tesxtstyle and 35dp for textsize.
- Using two EditText boxes to input task title and description. Set gravity central horizontal.
- Add a ListView and set the weight to 1 which let this listview occupy all the range except the form fields and the button
- We can custom a list layout which contains two textviews to show the title and description. Each TextView has been set gravity central to show texts in the middle of view. Also, we need a checkbox. The details will be mentioned later in graduate student's requirement.
- Set Layout margin of button view in order to don't touch the borders of the screen or the list. Also set gravity bottom right to place the buttom.

60 points:

Set an OnItemLongClickListener on ListView. If the LongClick happens, use adapter.getItem() and adapter.remove() functions to delete item from ListView.

80 points:

Add an onClick attribute on button in XML file and set an onClick listener in main activity. If the button has been clicked, get strings from EditText Views and call the adapter.add() function to add these strings in the format of Map<string,string> to the list. Also, we must use PrintStream object to put the current data into local txt file.

100 points:

We can override the OnResume() and OnPause() functions to achieve this goal. Saving the arraylist which stores the items in local txt file when OnPause() function called. Also, reading this txt file, load data into arraylist and create new adapter with this arraylist and set the adapter to listview when OnResume() function called. Thus, no matter which operations the user do, terminates or closes, the existing list will not be lost and can be loaded whenever reopen this application.

Graduate student's requirements see the next page

Graduate student’s requirement:

I custom my own list layout which contains two TextViews for title and description, I also add a CheckBox widget in this layout and put it at the right of TextViews. Thus, we finished the layout of single row of listview.

To make the checkbox work, we must custom list adapter. So we create a new class named CustomAdapter which extends ArrayAdapter. We load the row layout we customed in this adapter and set an onClick listener on checkbox.

I override getView() function of adapter. In this function, I write where to show title and description in single row via index parameter, which indicates the position or the index that the data item has in arrayList. So the adapter can transfer an arrayList to a ListView.

Also, I set an onClick listener on checkbox. If the checkbox has been clicked, we call the getItem() and remove() functions in ArrayAdapter class to remove the item whose checkbox has been clicked. At last, I use Handler class and postDelayed(new Runnable (),time) function to set 0.5s delay in order to show user that the checkbox has been clicked and then the item is deleted rather than delete this item immediately.