

Agenda

- passing Person data from first screen to second
- passing person object
- data tranfer from second to first
- data getting from second & sending it to third screen
- actionBar, toolBar
- menus & pop up menu
- creating menus using menu resource

App1 -> Transfer data from first to second Activity

- 1. Create a new Project with MainActivity
- 2. Add a new Activity called as InputActivity and make it a launcher activity
- 3. Design UI for InputActivity
- 4. Add the business logic inside InputActivity

```
btnsave.setOnClickListener(this);
btncancel.setOnClickListener(this);

@Override
public void onClick(View v) {
    if(v.getId()==R.id.btnsave){
        //todo
    }
    else if(v.getId() == R.id.btncancel){
        //todo
    }
}
```

- 5. Design UI for MainActivity
- 6. Add business logic inside MainActivity

App2 -> Transfer Person Object from first to second Activity

- 1.Copy the entire UI from APP1
- 2.Create a class called as Person.
- 3.Implement Serializable interface in person class
- 4.Create the person object and populate it with the data from edittext
- 5.Pass the person object from the intent to the next activity

App3 -> Pass the data From second activity to first activity

- 1. Create UI for MainActivity
- 2. Create UI for InputActivity
- 3. handle the click event of btngetData
- 4. Register your MainActivity for result

```
ActivityResultLauncher activityResultLauncher =
registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), new
ActivityResultCallback<ActivityResult>() {
    @Override
    public void onActivityResult(ActivityResult result) {
    }
});
```

- 5. launch the Input activity using the ActivityResultLauncher object

```
Intent intent = new Intent(this, InputActivity.class);
activityResultLauncher.launch(intent);
```

- 6. In InputActivity fetch the data from edittext and add it in the intent
- 7. Put that intent into the result object by calling a method setResult(0,intent)

```
Intent intent = new Intent();
intent.putExtra("name", name);
setResult(0, intent);
finish();
```

App4 -> Get the data from second activity and display it on third activity

- 1. Add 2 buttons getData and showData in activity_main.xml
- 2. Implement their onclick listeners in the Main activity
- 3. getData will go towards InputActivity to get the data of name and course
- 4. showdata will navigate towards DetailsActivity to display the data.
- 5. register your main activity for result to get the data from InputActivity
- 6. When the data is available in the activityresult call back method add the data in the MainActivity class field.

- 7. Use the data kept in these files to pass it through intent to the DetailsActivity.

ActionBar (App5)

- 1. If you want the action bar on all activities then change the themes.xml file.
- 2. Select the theme as Theme.Material3.DayNight into themes.xml file
- 3. If you don't want the action bar on specific activities then use.

```
getSupportActionBar().hide();
```

- 4. If you want to add the title and the back action to action bar then

```
getSupportActionBar().setTitle("Details");  
getSupportActionBar().setDisplayHomeAsUpEnabled(true);
```

- 5. To make the back action working add the parent attribute in manifest file

```
<activity  
    android:parentActivityName=".MainActivity"  
    android:name=".MainActivity2"  
    android:exported="false" />
```

ToolBar (App6)

- 1. add the toolbar into activity_main.xml

```
<androidx.appcompat.widget.Toolbar  
    android:id="@+id/toolbar"  
    android:background="@android:color/darker_gray"  
    app:title="MainActivity"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

- 2. If we want to add the menus to our toolbar then we need to set the behaviour of toolbar as actionBar

```
Toolbar toolbar = findViewById(R.id.toolbar);  
setSupportActionBar(toolbar);
```

- 3. To add the menus to the toolbar use the method `onCreateOptionsMenu()`

```
public boolean onCreateOptionsMenu(Menu menu) {
    menu.add("Profile");
    menu.add("Settings");
    menu.add("Logout");
    return super.onCreateOptionsMenu(menu);
}
```

- 4. To handle the click events of the menus from the toolbar use the `onOptionsItemSelected()`

```
public boolean onOptionsItemSelected(@NonNull MenuItem item) {
    if(item.getTitle().equals("Profile")){
        Intent intent = new Intent(this, ProfileActivity.class);
        startActivity(intent);
    }
    else if(item.getTitle().equals("Settings")){
        Intent intent = new Intent(this, SettingsActivity.class);
        startActivity(intent);
    }
    else if(item.getTitle().equals("Logout")){
        finish();
    }
    return super.onOptionsItemSelected(item);
}
```

Popup Menu (App6)

- To add the popup menu create the object of the popup menu and then add the menus that you want

```
PopupMenu popupMenu = new PopupMenu(this, view);
Menu menu = popupMenu.getMenu();
menu.add("Archive");
menu.add("Delete");
popupMenu.show();
```

- To handle the click events of popup menu implement the popup menu onclick listener

```
popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
    @Override
    public boolean onMenuItemClick(MenuItem item) {
        if(item.getTitle().equals("Archive"))
            Toast.makeText(MainActivity.this, "Archive is clicked",
```

```
Toast.LENGTH_SHORT).show();  
        else if(item.getTitle().equals("Delete"))  
            Toast.makeText(MainActivity.this, "Delete is clicked",  
Toast.LENGTH_SHORT).show();  
        return false;  
    }  
});
```