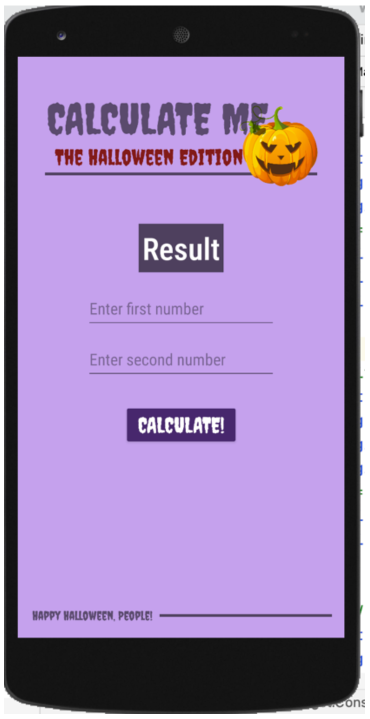
WIA2007 Mobile Application Development
Semester 1, Session 2022/2023
Practical 4 (User Interface: Layouts, Views and Widgets)

Pre-Task: Plan for Activity

Plan and draft for an activity in your individual app that utilizes the following elements:

- ☑ TextView
- ☑ EditText
- ☑ Button
- ☑ Checkbox

Task 1: The Addition Function



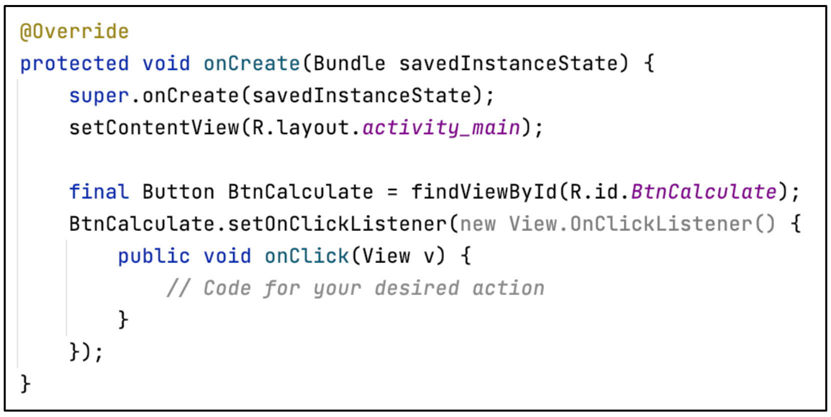
In this demonstration, we will use **two EditText**, **one Button** and **one TextView**.

The naming convention used in this demonstration is:

- ETNum1 – to input first number.
- ETNum2 – to input second number.
- BtnCalculate – to perform addition operation.
- TVResult – to display the result of the addition operation.

In this practical, we will learn how to create “action” to the button. For our addition function here, the “action” is when the button is clicked, the mobile app will sum the first number and the second number input by user from ETNum1 and ETNum2.

In your .java file, specify the following codes to enable onClick “action” to your button.

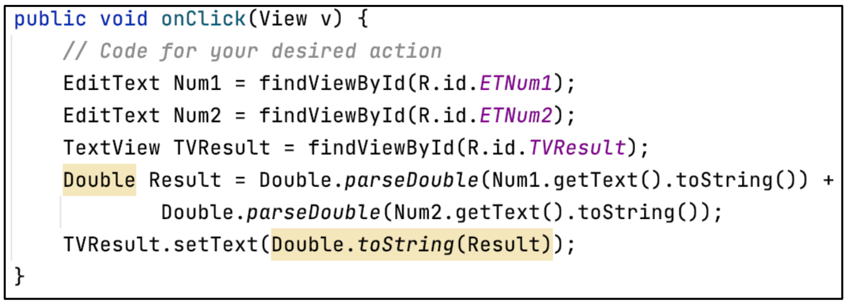


```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    final Button BtnCalculate = findViewById(R.id.BtnCalculate);
    BtnCalculate.setOnClickListener(new View.OnClickListener() {
        public void onClick(View v) {
            // Code for your desired action
        }
    });
}
```

The java code above will create an instance of View.OnClickListener and link it to the button using setOnClickListener(View.OnClickListener). Later on, when the button is clicked, the system will execute the code in onClick(View).

To perform the summation function, use the following code:



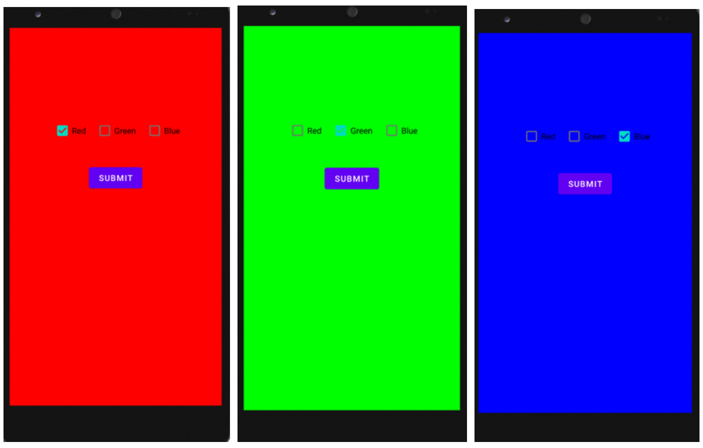
```java
public void onClick(View v) {
    // Code for your desired action
    EditText Num1 = findViewById(R.id.ETNum1);
    EditText Num2 = findViewById(R.id.ETNum2);
    TextView TVResult = findViewById(R.id.TVResult);
    Double Result = Double.parseDouble(Num1.getText().toString()) +
            Double.parseDouble(Num2.getText().toString());
    TVResult.setText(Double.toString(Result));
}
```

Now try to run your code to see whether you are able to sum the two numbers or not. *What are the other improvements that you can suggest to enhance the usability of the codes?*

Besides, there are many methods to create onClick action to your button. *What are the other approaches?*

Task 2: Color Selection

In this task, we will need **three checkboxes** (named as CBRed, CBGreen and CBBlue) and **one button** (BtnSubmit).



When the Submit button is clicked, the system will run the following codes:

```java
public void BtnSubmitOnClick (View v){
    CheckBox CBRed = findViewById(R.id.CBRed);
    CheckBox CBGreen = findViewById(R.id.CBGreen);
    CheckBox CBBlue = findViewById(R.id.CBBlue);

    ConstraintLayout CSLayout = findViewById(R.id.CLayoutCS);

    if(CBRed.isChecked())
        CSLayout.setBackgroundColor(Color.RED);
    else if(CBGreen.isChecked())
        CSLayout.setBackgroundColor(Color.GREEN);
    else
        CSLayout.setBackgroundColor(Color.BLUE);
}
```

The codes will check which checkbox is checked (returns true if checked, false otherwise), then it will change the background of the layout accordingly.

Do you think that the codes are doing the perfect job? What are the possible problems of this codes? Fix the problem using your creativity!

Submission

Submit the screenshot of your individual mobile app that uses EditText, TextView, Button and CheckBox components to Spectrum.