Termwork 1:

1. onCreate(Bundle savedInstanceState):
    - This is the entry point of the activity.
    - It initializes the activity, sets the content view from the layout resource (R.layout.activity_main), and retrieves references to the UI elements.
2. setContentView(int layoutResID):
    - It sets the activity content from a layout resource.
3. findViewById(int id):
    - It is used to find a view that was identified by the id attribute from the XML layout.
    - In this case, it finds references to the TextView and two Button widgets.
4. setOnClickListener(View.OnClickListener listener):
    - Sets a click listener for the buttons to respond to user interaction.
5. onClick(View v):
    - This method is implemented for both buttons to define what happens when each button is clicked.
6. t.setTextSize(float size):
    - Sets the text size of the TextView to the specified value (font variable in this case).
7. t.setTextColor(int color):
    - Sets the text color of the TextView to the specified color.
    - The color is determined based on the value of the ch variable, which is incremented on each button click.
8. Color.RED, Color.GREEN, Color.BLUE, Color.CYAN, Color.YELLOW, Color.MAGENTA:
    - Constants representing various colors in the Android graphics Color class.
    - Used to set the text color of the TextView based on the value of the ch variable.
9. super.onCreate(savedInstanceState):
    - Calls the onCreate method of the superclass (AppCompatActivity).
    - Ensures that the essential setup for the activity is performed.
10. float font = 30; int ch = 1;:

    Member variables to store the font size and color change state.
11. if (font == 50) font = 30;:
    - Resets the font size to 30 if it reaches 50.
12. if (ch == 7) ch = 1;:
    - Resets the color change state to 1 if it reaches 7.

Termwork 2

1. onCreate(Bundle savedInstanceState):
    - This is a lifecycle method in Android activities.
    - It is called when the activity is first created.
    - It initializes the activity, sets the content view, and sets up the UI components.
2. findViewById(int id):
    - Used to find the View by its ID, which is specified in the XML layout file.
    - Returns the View that corresponds to the given ID.
3. ArrayAdapter(Context context, int resource, T[] objects):
    - Creates an ArrayAdapter for the Spinner.
    - It is used to adapt an array of data for the Spinner, in this case, the dept_array of department names.
4. setAdapter(Adapter adapter):
    - Sets the adapter for the Spinner to the one created using the ArrayAdapter.
5. setOnClickListener(View.OnClickListener l):
    - Sets the OnClickListener for the Button.
    - Defines the behavior that should occur when the Button is clicked.
6. onClick(View v) (inside the OnClickListener):
    - This method is called when the associated Button is clicked.
    - It retrieves values from EditText and Spinner, creates an Intent, and starts a new activity (SecondActivity).
7. getResources():
    - Retrieves the resources (like strings, drawables, etc.) for the application.
8. startActivity(Intent intent):
    - Starts the specified activity.
9. putExtra(String name, String value):
    - Adds extended data to the intent. In this case, it's used to pass data from the current activity to the next one.
10. setContentView(int layoutResID):
    - Sets the activity content to an explicit view.
    - In this case, it sets the content view to the layout specified in the XML file (activity_main.xml).
11. onBackPressed() (not explicitly used):
    - A method called when the back button is pressed.
    - It's part of the activity lifecycle and can be overridden to define custom behavior.

12. **getIntent() Method:**
    - Retrieves the intent that started this activity.
13. **getStringExtra(String name) Method:**
    - Retrieves extended data from the intent. In this case, it's used to get the values associated with the keys "name_key," "reg_key," and "dept_key."
14. **Variable Declarations:**
    - **TextView t1, t2, t3**: Declare three **TextView** variables to hold references to text view elements in the layout.
    - **String name, reg, dept**: Declare three String variables to store data received from the intent.
15. **Variable Initialization:**
    - Initializes the **TextView** variables by finding views with their respective IDs.
    - Initializes the **Intent** variable to get the intent that started this activity.
16. **Data Retrieval from Intent:**
    - Retrieves the string values associated with the keys "name_key," "reg_key," and "dept_key" from the intent.
17. **Data Setting to TextViews:**
    - Sets the retrieved values to the **TextView** elements (**t1**, **t2**, **t3**) in the layout.

Termwork 3:

1. onCreate(Bundle savedInstanceState) method:
    - This method is part of the Android Activity lifecycle and is called when the activity is first created.
    - It initializes the user interface, sets the content view using setContentView(R.layout.activity_main), and references various UI elements like EditText, Button, and TextView.
2. UI Element References:
    - Num1, Num2: References to the two EditText views for user input.
    - Add, Sub, Mul, Div: References to the four Button views for addition, subtraction, multiplication, and division operations.
    - Result: Reference to the TextView where the calculation result will be displayed.
3. onClick(View v) method:
    - This method is part of the OnClickListener interface and is implemented to handle button clicks.
    - It checks if the input fields are not empty, retrieves the numerical values from Num1 and Num2, and performs the corresponding arithmetic operation based on the button clicked.
    - The result is then displayed in the Result TextView.
4. Arithmetic Operations:
    - Addition (+), Subtraction (-), Multiplication (*), and Division (/) operations are performed based on the button clicked in the onClick method.
    - The results are calculated and stored in the result variable.
5. Validation:
    - The code checks if either of the input fields (Num1 or Num2) is empty before proceeding with the calculations.
6. Type Conversion:
    - The values entered in the EditText fields are retrieved as strings and then converted to floating-point numbers using Float.parseFloat().
7. Switch Statement:
    - The switch statement is used to determine which button was clicked and perform the corresponding arithmetic operation.
8. Setting Result Text:
    - The final result is displayed in the Result TextView by setting its text property with the formatted result string.

Termwork 4:

1. onCreate(Bundle savedInstanceState):
    - This method is part of the Android Activity lifecycle and is called when the activity is first created.
    - It initializes the UI components, such as EditText fields and buttons, and sets the content view to the layout defined in "activity_main.xml."
    - It also establishes a connection to the SQLite database, creating a table named "student" if it does not exist.
2. onClick(View view):
    - This method implements the OnClickListener interface, handling click events for various buttons.
    - It performs different database operations based on the button clicked (Insert, Delete, Update, View, ViewAll).
    - It validates input data, executes SQL queries, and displays success or error messages using the showMessage() method.
3. showMessage(String title, String message):
    - This method creates and shows an AlertDialog with the specified title and message.
    - AlertDialogs are used to display informative messages and prompts to the user.
    - It utilizes the Builder class to construct and customize the dialog before displaying it.
4. clearText():
    - This method clears the text in the Rollno, Name, and Marks EditText fields.
    - It also sets the focus on the Rollno field, providing a clean slate for the user to input new data.
5. db.execSQL(String sql):
    - Executes the provided SQL statement, which is used for creating the "student" table, inserting, updating, or deleting records.
    - It's a method of SQLiteDatabase class, allowing direct execution of SQL commands without returning any result set.
6. openOrCreateDatabase(String name, int mode, CursorFactory factory):
    - Opens or creates a database with the specified name, mode, and cursor factory.
    - This method is used to establish a connection to the SQLite database, providing the foundation for executing SQL commands.
7. Cursor db.rawQuery(String sql, String[] selectionArgs):
    - Executes a raw SQL query and returns a Cursor over the result set.
    - Used for retrieving data from the database based on the provided SQL query, such as selecting, updating, or deleting records.
8. Cursor.moveToFirst():
    - Moves the cursor to the first row of the result set.
    - It is often used to check if a record exists in the result set before performing operations like deletion or updating.

9. EditText.getText():

    - Retrieves the text entered in an EditText field.
    - Used to obtain the values entered by the user for Rollno, Name, and Marks fields during various database operations.

10. StringBuffer:

- A mutable sequence of characters. In this context, it is used to build a formatted string containing student details for display in the AlertDialog.
- The buffer is appended with information for each student record retrieved from the database.

Termwork 5

1. onCreate(Bundle savedInstanceState):
    - This method is part of the Android Activity lifecycle and is called when the activity is first created.
    - It initializes the activity's user interface using the layout specified in activity_main.xml.
2. notify.setOnClickListener(new View.OnClickListener()):
    - Sets an OnClickListener on the notify button, which means the specified code block will be executed when the button is clicked.
3. Intent intent = new Intent(MainActivity.this, SecondActivity.class):
    - Creates an explicit intent to launch the SecondActivity when the button is clicked.
4. PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0):
    - Creates a pending intent that wraps the intent to start the SecondActivity. It is used for deferred actions, in this case, starting the second activity when the notification is clicked.
5. Notification.Builder(MainActivity.this):
    - Creates a builder for building a Notification object with specified attributes.
6. setSmallIcon(R.mipmap.ic_launcher):
    - Sets the small icon for the notification. It uses the launcher icon (ic_launcher) as the small icon.
7. setContentTitle("New Message"):
    - Sets the title for the notification.
8. setContentText(e.getText().toString()):
    - Sets the text/content of the notification. It retrieves the text from an EditText field (e) and converts it to a string.
9. noti.flags |= Notification.FLAG_AUTO_CANCEL:
    - Sets the FLAG_AUTO_CANCEL flag for the notification, which automatically cancels the notification when the user clicks it.
10. manager.notify(0, noti):
    - Sends the notification to the NotificationManager with a unique identifier (0 in this case).