Name: Ang Sengleap
Class: A4

Explain each code.

main() Function

```
void main() {
  runApp(const MyApp());
}
```

The main() function is the entry point of the Flutter application. It calls runApp(), which inflates the widget and attaches it to the screen.

MyApp Widget

```
class MyApp extends StatelessWidget {
  const MyApp({super.key});


  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home: const HomePage(),
    );
  }
}}
```

MyApp is a StatelessWidget that returns a MaterialApp. MaterialApp provides material design structure, routing, and theming. The HomePage widget is set as the home screen.

HomePage Widget

```
class HomePage extends StatefulWidget {
  const HomePage({super.key});

  @override
  State<HomePage> createState() => _HomePageState();
}
```

HomePage is a StatefulWidget because the selected tab can change. It manages the BottomNavigationBar and multiple navigation stacks.

Current Index Variable

```
int _currentIndex = 0;
```

The _currentIndex variable stores the index of the currently selected tab. It determines which tab content is visible.

Navigator Keys

```
final List<GlobalKey<NavigatorState>> _navigatorKeys = [
  GlobalKey<NavigatorState>(),
  GlobalKey<NavigatorState>(),
  GlobalKey<NavigatorState>(),
];
```

GlobalKey objects are used to uniquely identify each Navigator. Each tab has its own Navigator key, allowing it to maintain a separate navigation stack.

onTap Method

```
void _onTap(int index) {
  if (_currentIndex == index) {
    _navigatorKeys[index]
        .currentState!
        .popUntil((route) => route.isFirst);
  } else {
    setState(() => _currentIndex = index);
  }
}}
```

The _onTap method handles tab selection. If the same tab is tapped again, the navigation stack is popped back to the first screen. If a new tab is selected, the UI updates accordingly.

WillPopScope

```
return WillPopScope(
  onWillPop: () async {
    final isFirstRouteInCurrentTab =
        !await _navigatorKeys[_currentIndex]
            .currentState!
            .maybePop();
    return isFirstRouteInCurrentTab;
  },
```

WillPopScope intercepts the system back button. If the current tab has pages in its stack, it pops them instead of closing the app.

IndexedStack

```
body: IndexedStack(
  index: _currentIndex,
  children: [
    _buildNavigator(_navigatorKeys[0], const TabOne()),
    _buildNavigator(_navigatorKeys[1], const TabTwo()),
    _buildNavigator(_navigatorKeys[2], const TabThree()),
  ],
),
```

IndexedStack keeps all tab widgets alive while displaying only the selected one. This ensures that navigation history is preserved when switching tabs.

Navigator Widget

```
Widget _buildNavigator(
    GlobalKey<NavigatorState> key, Widget child) {
  return Navigator(
    key: key,
    onGenerateRoute: (settings) {
      return MaterialPageRoute(
        builder: (context) => child,
      );
    },
  );
}
```

Each tab contains its own Navigator widget. The Navigator manages a stack of routes (pages) independently for each tab.

BottomNavigationBar

```
bottomNavigationBar: BottomNavigationBar(
  currentIndex: _currentIndex,
  onTap: _onTap,
  items: const [
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      label: 'Home',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.search),
      label: 'Search',
    ),
    BottomNavigationBarItem(
      icon: Icon(Icons.person),
      label: 'Profile',
    ),
  ],
),
```

BottomNavigationBar displays tabs at the bottom of the screen. Each item represents a tab, and selecting one changes the visible content.

Tab Screens

```
class TabOne extends StatelessWidget {
  const TabOne({super.key});


  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Home Tab')),
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(
                builder: (_) => const DetailPage(title: 'Home Details'),
              ),
            );
          },
          child: const Text('Go to Details'),
        ),
      ),
    );
  }
}
```

Each tab screen is a StatelessWidget with its own Scaffold. Navigator.push() is used to navigate to detail pages within the same tab.

Detail Page

```
class DetailPage extends StatelessWidget {
  final String title;

  const DetailPage({super.key, required this.title});
```

The DetailPage widget represents a secondary screen. It demonstrates how navigation works inside a tab without affecting others.