Week - 1 Quiz Solutions

1. What is the purpose of using a Tab Layout in an App?
   Ans: Provide a Horizontal Layout to display tabs

2. What is the name of the container that holds the tabs in Jetpack Compose?
   Ans: TabRow

3. What is the role of the indicator in Tab Layout?
   Ans: Display which tab is selected currently

4. Which of the following is not a type of Floating Action Button?
   Ans: Normal

5. Which parameter of the Floating Action Button is used to invoke functionality?
   Ans: OnClick

The code for the task is also given in the Week -1 folder in the GitHub repository.

The only purpose of this code is to view the composable Main() function so that we can view it while running the app.

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            New1Theme {
                Main()
            }
        }
    }
}
```

The snippet of the code given for the task has been completed here. For implementing the pager, we have made use of a module called the accompanist pager. This needs to be added in your build.gradle.kts file (screenshot given below). Once you have implemented it and synced gradle, the pager setup is ready.

```
@OptIn(ExperimentalPagerApi::class)
@Composable
fun Main() {
    val titles = listOf("Food", "Fun")
    val pagerState = rememberPagerState()
    val scope = rememberCoroutineScope()

    Column(modifier = Modifier.fillMaxSize()) {
        TabRow(
            selectedTabIndex = pagerState.currentPage,
            backgroundColor = Color(116, 201, 49),
            indicator = { tabPositions ->
                TabRowDefaults.Indicator(
                    modifier = Modifier.pagerTabIndicatorOffset(pagerState, tabPositions),
                    height = 4.dp,
                    color = Color.Black
                )
            }
        ) {
            titles.forEachIndexed { index, title ->
                androidx.compose.material3.Tab(
                    selected = pagerState.currentPage == index,
                    onClick = {
                        scope.launch {
                            pagerState.animateScrollToPage(index)
                        }
                    },
                    text = { Text(title) }
                )
            }
        }
        HorizontalPager(
            count = titles.size,
            state = pagerState,
            modifier = Modifier.weight(1f)
        ) { page ->
            when (page) {
                0 -> FoodScreen()
                1 -> FunScreen()
            }
        }
    }
}
```

build.gradle.kts (Module: app)

```
implementation("com.google.accompanist:accompanist-pager:0.27.1")
implementation("com.google.accompanist:accompanist-pager-indicators:0.27.1")
```

Then, we need to implement the floating action button. For now, we aren't going to add any functionality to the button but just design the UI in the Food Screen.

```
@Composable
fun FoodScreen() {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .padding(20.dp),
        verticalArrangement = Arrangement.Bottom,
        horizontalAlignment = Alignment.End,
    ) {
        androidx.compose.material.FloatingActionButton(
            onClick = {
                //to be implemented
            },
            backgroundColor = Color(116, 201, 49),
            elevation = FloatingActionButtonDefaults.elevation(
                defaultElevation = 20.dp
            )
        ) {
            Icon(Icons.Filled.Add, "Floating action button.")
        }
    }
}
```

It is important to note that this isn't the only way to build the app. There are many ways in which you can choose to proceed and this is just one of them.