BCS 371 Mobile Application Development I

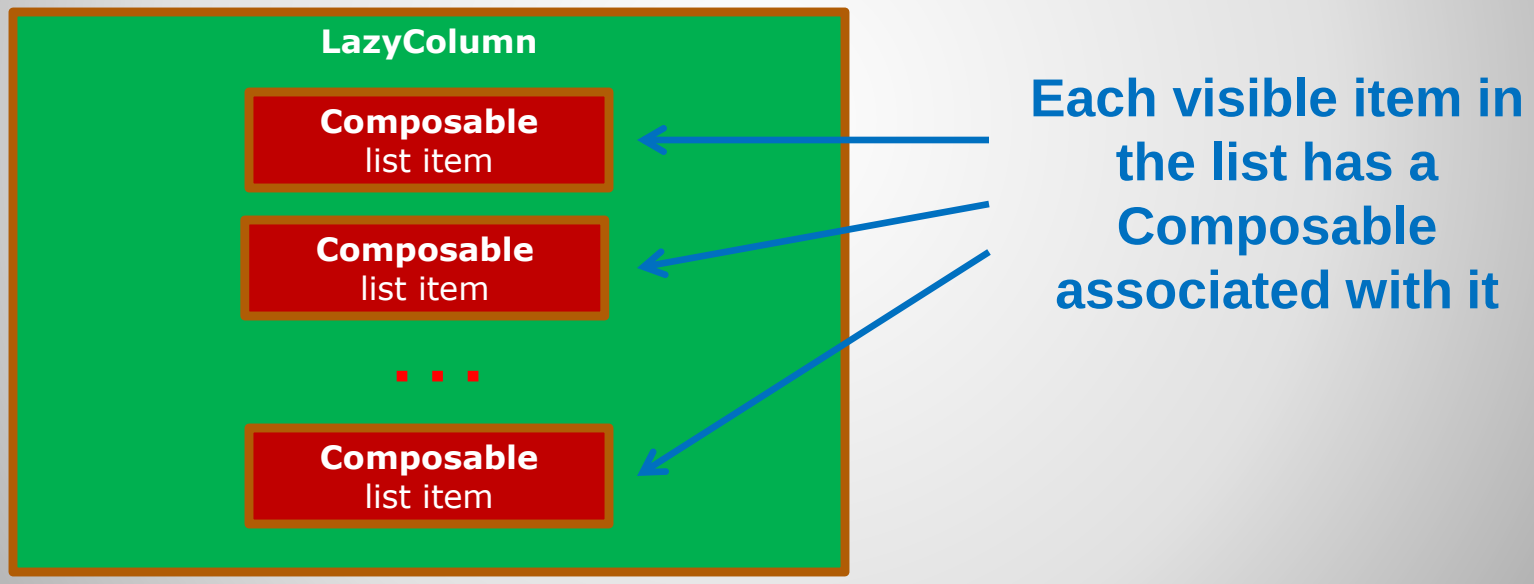
Arthur Hoskey, Ph.D.
Farmingdale State College
Computer Systems Department

Today's Lecture

- LazyColumn

LazyColumn

- Used to display a list of items **EFFICIENTLY**.
- Can be tricky to set up, though.



LazyColumn

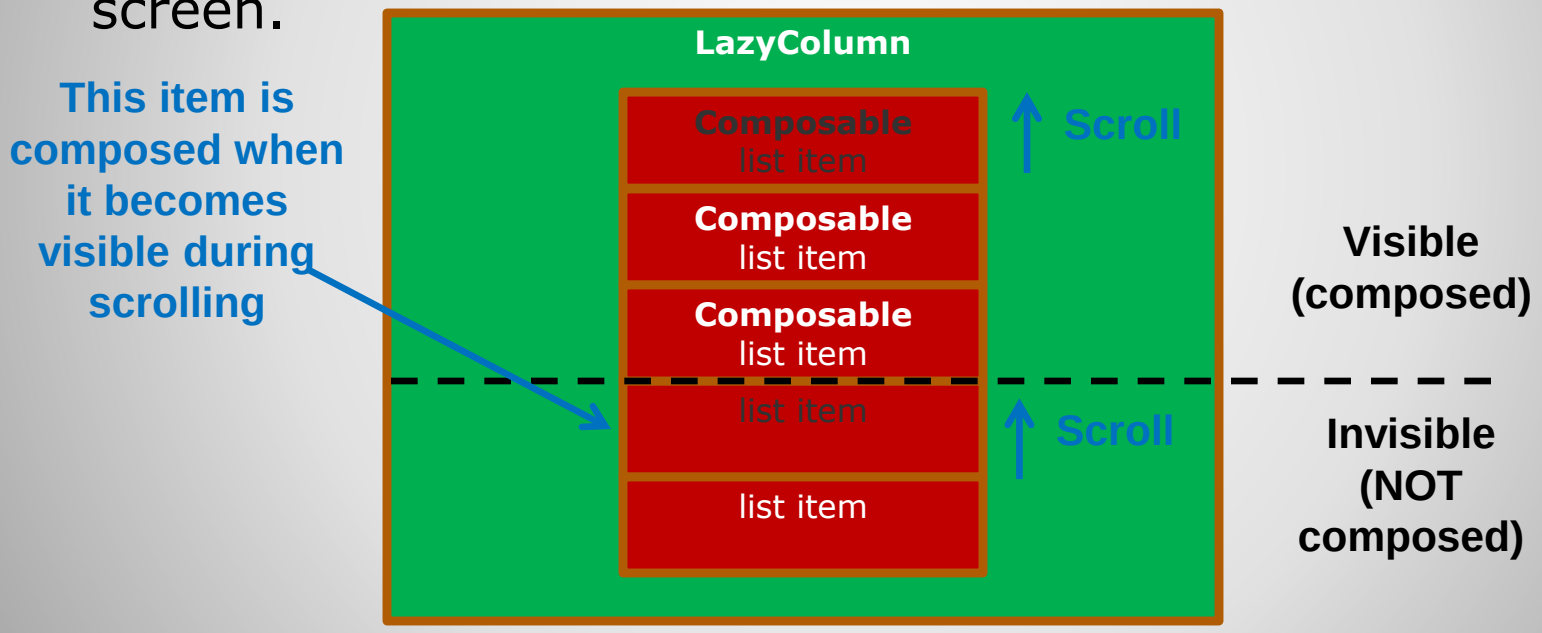
LazyColumn Efficiency

- **Compostion Efficiency.** Only composes list items that are **visible** in the GUI at the present time.
  - No reason to compose items that are not visible.
  - For example, if the list contained 1000 items but only 10 were visible it would be a waste of CPU time to compose all 1000 items (the user may never scroll to them).
  - "Lazy Evaluation". Put off doing something until you absolutely must do it.

LazyColumn Efficiency

LazyColumn Efficiency

- When the user scrolls, the first item will be moved off screen and the fourth item will be moved on screen.
- Initially, only the first 3 items are composed.
- The fourth item is composed only when it appears on screen.



LazyColumn Efficiency

LazyColumn Setup

- LazyColumn needs to be given a collection to iterate over.
- As it visits each item, it will display data for that item.

The items function is passed a collection. It will iterate over all items in the given collection (collectionToDisplay is a variable declared elsewhere that contains a collection of data)

currentItem is the lambda variable. currentItem contains the current item being iterated over.

```
LazyColumn {

    items(collectionToDisplay) {currentItem ->
        // Code to display one item goes here (Text etc…)
    }
}
```

Use currentItem variable in the display code

LazyColumn Setup

LazyColumn Example

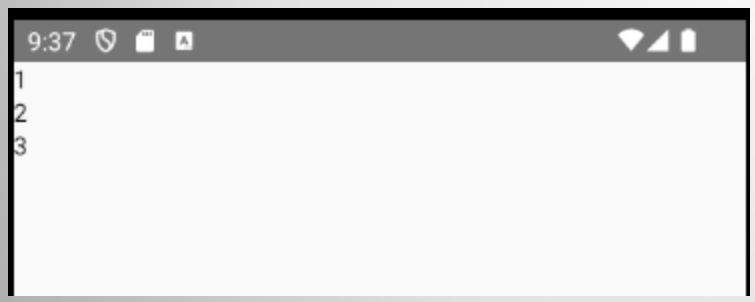
```
var numList = mutableListOf(1, 2, 3)
LazyColumn {
    items(numList) { currentItem ->
        Text(text = currentItem.toString())
    }
}
```

Create a list of numbers to display

items takes a list. It will iterate over all items in the given list.

currentItem is the lambda variable. currentItem contains the value of the current item. The variable name does not have to be "currentItem". For example, you could use "x" as the variable name if you want.

Put the current number in a Text composable



LazyColumn Example

LazyColumn - Item Click

- Surround each item with a Box composable.
- Add a clickable modifier to the Box (handles the click event).
- Put click event on the Box. A click anywhere in the Box triggers a click event.

```
LazyColumn {
    var numList = mutableListOf(1, 2, 3)
    items(numList) { currentItem ->
        Box(
            modifier = Modifier
                .fillMaxSize()
                .clickable {
                    // Click event handler code goes here…
                }
        ) {
            Column {
                Text(text = currentItem.toString())
                // Other item composables can go here…
            }
        }
    }
}
```

Surround the item with a Box

Makes it handle clicks anywhere in the Box

Click event handler code goes here

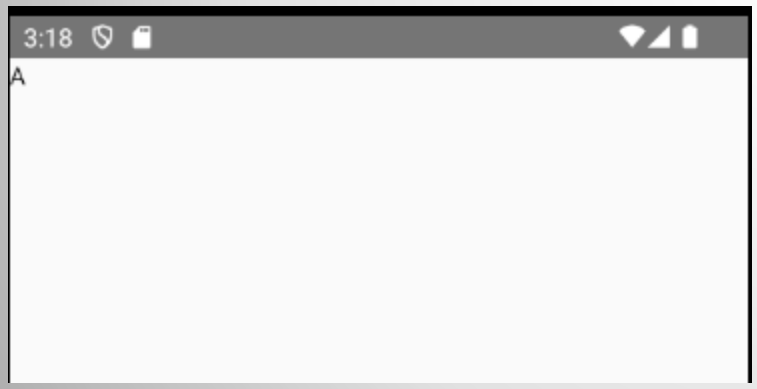
Multiple composables that make up an item go here (inside the Box)

LazyColumn - Item Click

LazyColumn – Add Single Item

```
LazyColumn {

    item {
        Text("A")
    }

}
```

Use the item function (instead of items) to show a single item in the LazyColumn



LazyColumn – Add Single Item

LazyColumn - Multiple Functions

- The item and items functions can be called multiple times.

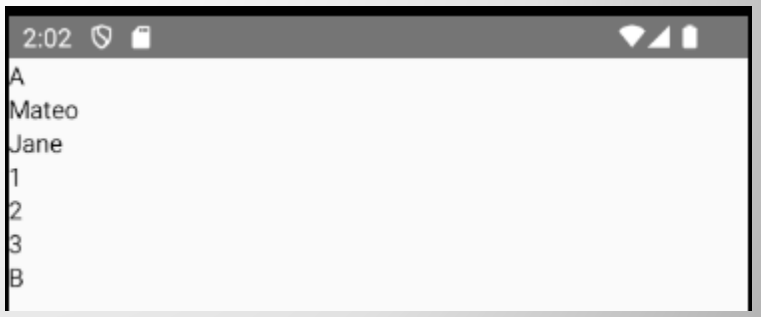
```
var nameList = mutableListOf("Mateo", "Jane")
var numList = mutableListOf(1, 2, 3)

LazyColumn {
    item {
        Text("A")
    }
    items(nameList) {currentItem ->
        Text(currentItem)
    }
    items(numList) {currentItem ->
        Text(currentItem.toString())
    }
    item {
        Text("B")
    }
}
```

The item and items functions are called twice within the LazyColumn

Single item

Different collections are being displayed in each call to items (nameList and numList)

Single item



LazyColumn – Multiple Functions

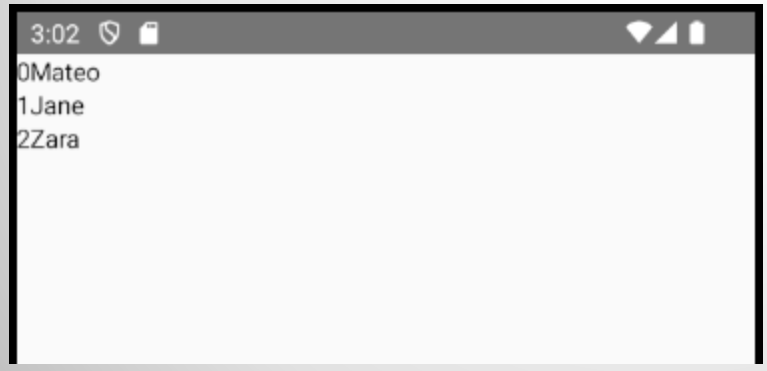
LazyColumn - itemsIndexed

- The itemsIndexed function also provides the index of the current item.

```
var nameList = mutableListOf("Mateo", "Jane", "Zara")
LazyColumn {

    itemsIndexed(nameList) { currentIndex, currentItem ->
        Row {
            Text(text = currentIndex.toString())
            Text(text = currentItem.toString())
        }
    }
}
```

Two variables. One for the current index and one for the current item.

This displays both the index and the item (both in the same row).



Each row shows the index and the item data

LazyColumn – itemsIndexed

- End of Slides

End of Slides