Your grade: 80%

Your latest: **80%** • Your highest: **80%** • To pass you need at least 80%. We keep your highest score.

Next item →

1. What is the back stack used for in Android? **1 / 1 point**

- (●) backward navigation
- ( ) generating routes
- ( ) database queries
- ( ) forward navigation

Correct
Correct! Upon launching the application, the start destination is placed into the stack. When another destination is opened it will be placed on top of the stack.

2. The type of the `route` variable which corresponds to the destination in the navigation graph is _____ . **1 / 1 point**

- ( ) `Int`
- (●) `String`
- ( ) `Boolean`

Correct
Correct! The `route` is a `String` unique for every destination.

3. Which of the following is the correct way to create a destination object for a Profile screen that implements the below `Destinations` interface? **1 / 1 point**

```
interface Destinations{
 val route:String
}
```

( )

```
Profile:Destinations{
    override val route = “Profile”
}
```

( )

```
object Profile{
    override val route = “Profile”
}
```

( )

```
object Destinations{
    override val route = “Profile”
}
```

(●)

```
object Profile:Destinations{
    override val route = “Profile”
}
```

Correct
Correct! The object keyword is used to create the object without creating the class. The object correctly extends the `Destinations` interface.

4. Which of the following data types are supported by `Navigation`? Select three that apply. 1 / 1 point

- [x] Integer

Correct
Correct! Integer types are defined as `NavType.IntType`.

- [x] Boolean

Correct
Correct! Boolean types are defined as `NavType.BoolType`.

- [x] String

Correct
Correct! String is the default type when the type is not specified.

- [ ] Array

5. Given the below function `ProfileScreen`, which of the following is the correct definition of its corresponding destination? 1 / 1 point

```
@Composable
fun ProfileScreen(userName: String?) {
    Column(
        Modifier.fillMaxSize(),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Welcome, $userName",
            fontSize = 33.sp
        )
    }
}
```

○

```
interface Destinations {
    val route: String
}
object Profile : Destinations {
    argUserName = "UserName"
    override val route = "Profile"
}
```

◉

```
interface Destinations {
    val route: String
}
object Profile : Destinations {
    const val argUserName = "UserName"
    override val route = "Profile"
}
```

○

```
interface Destinations {
    val route: String
}
object Profile{
    const val argUserName = "UserName"
    override val route = "Profile"
}
```

Correct

Correct! The `ProfileScreen` function accepts an argument to be displayed. The argument to be passed is defined in the `Destination` object.

6. Which of the following is used for the primary destinations which need to be accessed from anywhere within the app? **0 / 1 point**

- ○ `bottomNavigation`
- ◉ `Navigation` component
- ○ `TopAppBar`

Incorrect
Not quite. Please review the video Tabbed Navigation.

7. What among the following elements is scrollable without adding the modifier? Select all that apply. **1 / 1 point**

- ☐ `Row`
- ☐ `Column`
- ☑ `LazyColumn`

Correct
Correct! `LazyColumn` is scrollable without adding the modifier.

- ☑ `LazyRow`

Correct
Correct! `LazyRow` is scrollable without adding the modifier.

8. What is the most suitable element when you have an unknown or large number of items? Select all that apply. **1 / 1 point**

- ☐ `Column`
- ☑ `LazyColumn`

Correct
Correct! `LazyColumn` creates a vertical scrollable list.

- ☐ `Row`
- ☑ `LazyRow`

Correct
Correct! `LazyRow` creates a horizontal scrollable list.

9. Imagine you have defined the destination below and you want to use the variable `argProductNumber` in the `NavHost`. **0 / 1 point**

```
object Product : Destinations {
    const val argProductNumber = "ProductNumber"
    override val route = "Product"
}
```

Which of the following is the correct way to append the `argProductNumber` to the Product `route` within the `NavHost`?

- ○ `Product.route + "/{$Product.argProductNumber}`
- ◉ `Product.route + "/{$argProductNumber}`
- ○ `Product/{ProductNumber}`

Incorrect
Not quite. Please review the reading Navigation Examples.

10. Which of the following best describes what a `NavHost` is? **1 / 1 point**

- ◉ `NavHost` acts as a container for displaying the current destination.
- ○ `NavHost` is a collection of navigable destinations.
- ○ `NavHost` is responsible for keeping track of the back stack.

Correct

Correct! `NavHost` takes the `NavController` as an argument and associates it with the `NavGraph`.