**1.**

The **manifest** file contains:

Select all that apply.

**1 / 1 point**



Permissions



**Correct**

Correct! You must register information about application permissions in the manifest.



Fragments



Activities



**Correct**

Correct! You must register information about application activities in the **manifest**.



Composables

**2.**

True or False: In Compose, the hierarchy is built through composition.

**0 / 1 point**



True



False



**Incorrect**

Correct! A **composable**, as the name suggests, can be composed within another **composable**.

**3.**

What can a layout contain? Select all that apply.

**0 / 1 point**



Other layouts



**Correct**

Correct! Layouts may contain other layouts.



Screens



UI Elements



**Correct**

Correct! Layouts may contain simple UI elements.



Activities



**This should not be selected**

Not quite. Please review the video

[Layouts in Compose](https://www.coursera.org/teach/create-the-user-interface-android-studio/wOGmOscqEeyduw6ktL3Xvw/content/item/lecture/c9ing/video-subtitles" \t "_blank)

[](https://www.coursera.org/teach/create-the-user-interface-android-studio/wOGmOscqEeyduw6ktL3Xvw/content/item/lecture/c9ing/video-subtitles" \t "_blank).



A **modifier** is responsible for modifying **Activity** behavior.



Modifiers allow you to impart specific interactivity or behavior to the elements.



**Correct**

Correct! You can, for example, make a button or a text clickable using modifiers.



When augmenting appearance, modifiers can act as decorators.



**Correct**

Correct! Modifiers decorate or style the UI elements or composables. Modifiers can, for example, modify the size, position, spacing and color.

**5.**

The count variable is defined within a composable function in the example below. Which of the following describes the default behavior of a recall to the same function?

6

7

8

    )

}



@Composable fun Counter() { val count = 0 Button(text = "+", onClick = { count.value++ } ) }



Increment the **count** variable value only during the recomposition.



Reset the **count** variable to its initial value.



Increment the **count** variable value.



**Correct**

Correct! To retain the value, you have to use the **remember** function and **mutableStateOf** observable.

**6.**

Which of the following does each composable destination in the navigation graph correspond to?



**Screen**



**Activity**



**Destination**



**Route**



**Correct**

Correct! Each composable destination corresponds to a **Route** in the navigation graph. The **Route** is a unique **String** for every **destination**.

**7.**

Each item in a Grid is known as a:

**1 / 1 point**



Column



Cell



Row



Image



**Correct**

Correct! In Android, a Grid is a scrollable layout that displays the items in two dimensions. A grid consists of both Rows and Columns. Each item in a Grid is known as a Cell.

**8.**

True or False: **Theme** is represented as a collection of attributes.

**1 / 1 point**



True



False



**Correct**

Correct! **Theme** is represented as a collection of attributes such as color, button height or text size.

**9.**

What numeric system is used to represent the ARGB color format?

**F2500FF1**

**1 / 1 point**



Binary number system



Hexadecimal number system



Octal number system



Decimal number system



**Correct**

Correct! The colors are represented with a Hexadecimal number system. The first two characters represent the value of the Alpha channel. The first two characters also represent the value of the Alpha channel.

**10.**

What is the output of the following code?

3

4

5

6

7

8

9

10

} Column {

    AnimatedVisibility(visible = visible) {

    Text(text = "Hello, world!")

} Button(onClick = { visible = !visible }) {

    Text("Button ")

    }

}



var visible by remember { mutableStateOf(true) } Column { AnimatedVisibility(visible = visible) { Text(text = "Hello, world!") } Button(onClick = { visible = !visible }) { Text("Button ") } }



The first **Button** click will instantly show the **Text** and the following **Button** click will have no effect.



The first **Button** click will instantly hide the **Text** and the following **Button** click will instantly show the Text.



The first **Button** click will start the fade in animation of the **Text** and following **Button** click will start the fade out animation of the **Text**.



The first **Button** click will start the fade out animation of the **Text** and the following Button click will start fade in animation of the **Text**.



**Incorrect**

Not quite. Please review the video

[Animated visibility](https://www.coursera.org/teach/create-the-user-interface-android-studio/wOGmOscqEeyduw6ktL3Xvw/content/item/lecture/d1QYi/video-subtitles" \t "_blank)

[](https://www.coursera.org/teach/create-the-user-interface-android-studio/wOGmOscqEeyduw6ktL3Xvw/content/item/lecture/d1QYi/video-subtitles" \t "_blank).

**11.**

Which of the following is represented by an **Activity**?



the main program entry point



a UI pattern



a component



the application screen



**Correct**

Correct! An **Activity** is an application component that contains the user interface and represents the application screen.

**12.**

Which of the following is the name of the class used to represent a **view** in the traditional view system?



**ViewGroup**



**View**



**TextView**



**ImageView**



**Correct**

Correct! All classes that represent specific UI elements inherit from **View**.

**13.**

Which of the following lifecycle states does an **Activity** spend the most time in?

**1 / 1 point**



Started



Resumed



Paused



Created



**Correct**

Correct! When an **Activity** gains focus and is ready for user interaction it is in the Resumed state.

**14.**

Which of the following is passed to an **Activity** as a layout resource than will be rendered on the screen?

**1 / 1 point**



a Lambda



a view



an XML file



a composable



**Correct**

Correct! An XML file stores structured information about how the Android OS should draw the screen.

**15.**

Which of the following is the approach used when building a user interface with Views and XML?

**0 / 1 point**



hierarchical



imperative



declarative



logical



**Incorrect**

Not quite. Please review the reading

[Using Views and XML: the downside](https://www.coursera.org/learn/create-the-user-interface-android-studio/supplement/Y71oR/using-views-and-xml-the-downside" \t "_blank)

[](https://www.coursera.org/learn/create-the-user-interface-android-studio/supplement/Y71oR/using-views-and-xml-the-downside" \t "_blank).

**16.**

Which of the following is the process of initially generating the hierarchy of UI elements?

**1 / 1 point**



recomposition



composition



composable



initialization



**Correct**

Correct! Jetpack Compose can be used to build a hierarchical UI consisting of composable functions.

**17.**

Which of the following is an annotation that allows developers to easily create components that are modular and reusable?

**1 / 1 point**



**@Composable**



**@Component**



**@Kotlin**



**@Annotation**



**Correct**

Correct! Jetpack Compose can be used to build a hierarchical UI consisting of composable functions.

**18.**

Which of the following is an invisible UI element that acts as a container to other UI elements?

**1 / 1 point**



**composable**



**column**



**row**



**layout**



**Correct**

Correct! Layouts are guides in organizing and arranging other UI elements on the screen.

**19.**

Which of the following built-in layouts in Compose places composables vertically?

**1 / 1 point**



row



box



column



layout



**Correct**

Correct! For a row the main axis is the vertical axis, so you can set the vertical alignment.

**20.**

Which of the following is the correct way to use a modifier to give a Column a background color of black and fill the available space?

**1 / 1 point**



1

2

3

4

Column(

 Modifier.fillMaxSize().backgroundColor(color: 0XFF000000)

)



Column( Modifier.fillMaxSize().backgroundColor(color: 0XFF000000) )



3

)



Column( Modifier.fillMaxSize().background(Color(color: 0XFF000000) )



1

2

3

Column(

 modifier = Modifier.fillMaxSize().background(Color(color: 0XFF000000)

)



Column( modifier = Modifier.fillMaxSize().background(Color(color: 0XFF000000) )



**Correct**

Correct! Modifiers can be chained and are applied from left to right.

**21.**

Backgreound modifier accepts what kind of object as an arugment?



Color class



Color object



**Background.color**



**BackgroundColor**



**Correct**

Correct! The background modifier accepts a color object as an argument.

**22.**

Which of the following is the default event handler added to the **Button** composable by Jetpack Compose?



**onStart**



**onPress**



**onClick**



**onInput**



**Correct**

Correct! The **onClick** event handler will execute the specified code in response to the **Button** being clicked.

**23.**

Which of the following event modifiers is used to specify a function to be executed when a user clicks a specific composable?

**1 / 1 point**



**Modifier.selectable**



**Modifier.swipeable**



**Modifier.clickable**



**Modifier.draggable**



**Correct**

Correct! The clickable modifier can be used to trigger actions such as navigation or updating a composable’s state.

**24.**

Which of the following best describes what state is in programming?

**1 / 1 point**



Data of a program that changes over time.



Data used to recreate an activity.



Data saved in memory.



The location of data within a program.



**Correct**

Correct! State can also be described as the current condition of a program.

**25.**

Which of the following informs developers on how to use and organize individual UI elements and exposes patterns, themes and examples of UI designs and layouts at various levels?

**1 / 1 point**



A design system



User interfaces



Component hierarchies



Built-in composables



**Correct**

Correct! A design system is a collection of components which carry standard and reusable UI elements.

**26.**

Imagine you want to create the ability for the user to navigate between two screens, Home and Dashboard. Which of the following is the correct way to define destination objects that implement the below interface?

3

}



interface Destinations{     val route:String }

**0 / 1 point**



1

2

3

4

5

6

object Home:Destinations{

    override val route = “Home”

}

object Dashboard:Destinations{

    override val route = “Dashboard”

}



object Home:Destinations{     override val route = “Home” } object Dashboard:Destinations{     override val route = “Dashboard” }



1

2

3

4

5

6

7

Home:Destinations{

    override val route = “Home”

}

Dashboard:Destinations{

    override val route = “Dashboard”

}



Home:Destinations{     override val route = “Home” } Dashboard:Destinations{     override val route = “Dashboard” }



1

2

3

4

5

6

object Home:Destinations{

    val route = “Home”

}

object Dashboard:Destinations{

    val route = “Dashboard”

}



object Home:Destinations{     val route = “Home” } object Dashboard:Destinations{     val route = “Dashboard” }



Not quite. Please review the video

[Navigation in Compose](https://www.coursera.org/learn/create-the-user-interface-android-studio/lecture/htXEr/navigation-in-compose" \t "_blank)

[](https://www.coursera.org/learn/create-the-user-interface-android-studio/lecture/htXEr/navigation-in-compose" \t "_blank).

**27.**

Which of the following Jetpack Compose layout arrangements adds equal space to all children?



space around



start



space evenly



center



**Correct**

Correct! The space evenly arrangement adds equal space to the children of a **List** including space at the start and end.

**28.**

Which of the following would you use to create a vertical scrollable **List** for a large or unknown number of items?

**1 / 1 point**



lazy row



lazy grid



lazy list



lazy column



**Correct**

Correct! Lazy column takes care of scrolling and recomposing items when required.

**29.**

Which of the following is a mechanism for unifying and maintaining the application look and feel across all screens?

**1 / 1 point**



a composable hierarchy



a theme



a design system



XML



**Correct**

Correct! The appearance of multiple screens can be controlled by a single theme.

**30.**

Which of the following APIs can be used to animate a value in Jetpack Compose?

**1 / 1 point**



remember API



animate API



animatedVisibility API



animate\*AsState API



**Correct**

Correct! the animate\*AsState API can animate several value types such as float, color and dp.