Quizlet

Study

FIT2081 chap 4,5,6

Terms in this set (13)

Why is the datatype of the input parameter declared as 'View'?	View data type can be any view data passed to the View class itself (string, int from UI components such as button or text) as they are all subclass to the View class itself.
a) What is the main difference between Guidelines and Barriers? b) When does the horizontal bias in constraint layout become redundant?	a) Barriers: barriers is a virtual wall that does not define its own position. Rather, it is connected to the largest size widget inside the barrier. Barrier's position is therefore flexible. Guideline: guideline is a virtual wall that does define its own position by showing the distance from start or end side of the layout to the respective guideline. Its position is fixed b) Horizontal bias in constraint layout become redundant when the layout size of a widget, for example, is set to match constraint mode. Match Constraint mode means the constraint will expands to reach the parent's layout. For example, horizontal match constraints will expand the size of the widget to constraint on both the start and end side of the parents layout. This eliminates the chance for that widget to move left or right

FIT2081 chap 4,5,6

a) What is the role of the second input parameter to BroadcastReceiver onRecieve() method?

b) If you have the following:
•Intent Filter ="week4.fit2081.quiz" · Key ="Key" · Data="FIT2081" Using the details above, write a piece of code that sends the Data in a broadcast message.

to request actions. Broadcast Receiver's roles is to receive intent as part of its input parameter and it will take into account if the broadcast is registered with intent filter to invoke corresponding correct method to the intent.

b)
public static final String WEEK4.FIT2081.QUIZ= "WEEK4.FIT2081.QUIZ";
public static final String KEY= "KEY";
Intent intent1=new Intent();
intent1.setAction(WEEK4.FIT2081.QUIZ);
intent1.putExtra(KEY,"FIT2081");
context.sendBroadCast(intent1);

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

FIT2081 chap 4,5,6

Code the missing line that allows the 'onReceive' method in the activity above to get invoked each time the following piece of code gets executed.	`savedInstanceState){`
	
	`.`
	
	`registerReceiver(new BroadcastHandler(),new IntentFilter("QuizFilter"));`
	
	`}`
	
	`@Override`
	`public void onReceive(Context context, Intent intent){`
	`String data1=intent.getStringExtra("Key");`
	`Toast.makeText(context: this, message:data1,`
	`Toast.LENGTH_LONG).show();`
	`}`

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

FIT2081 chap 4,5,6

Differentiate between views, viewGroup, layout, layout.Paramts

ViewGroup objects.
Layout is responsible for managing the size, position, and behaviour of all the View it containst

View class – Represents the basic building block for UI components. A View occupies the area on screen responsible for drawing and event handling. Is the base class for widgets, which are used to create interactive UI components (buttons, text fields etc).

ViewGroup subclass – The base class for Layouts and Views containers, it is a special View that contains other View's as its children. A ViewGroup allows View's to be nested, which is represented as a hierarchy called a Layout.

ViewGroup.LayoutParams – Sets of XML attributes used by the parent ViewGroup to define the size and position to each child View

Every Layout Class contains:
1. XML attributes (also inherit attributes from View and ViewGroup)

2. Java methods which applies directly to the View's Layout Params object e.g. addView(View child)

FIT2081 chap 4,5,6

sendBroadcast(), sendStickyBroadcast() or sendOrderedBroadcast() methods of the Activity class (the latter being used when results are required from the broadcast). In addition to providing a **messaging** and **event system** between application components, broadcast intents are also used by the Android system to notify interested applications about key system events (such as the external power supply or headphones being connected or disconnected).[1]

example code:
Intent intent = new Intent();intent.setAction("intent.filter.data");
intent.putExtra("key", "SomeData");
sendBroadcast(intent);

Broadcast receiver:
Base class for code that receives and handles broadcast intents

Step: Override onReceive:
class MyBroadCastReceiver extends BroadcastReceiver { @Override public void onReceive (Context context,Intent intent) { } }

Create instance of broadcast receiver:
MyBroadCastReceiver myBroadCastReceiver = new MyBroadCastReceiver();

Register broadcast receiver:
registerReceiver(myBroadCastReceiver,new IntentFilter(SMS_key, smsfilter))-key,value

Broadcast intent/broadcast receiver

FIT2081 chap 4,5,6

Differentiate between styles, themes and material design	format for a View or window. A style is defined in an XML resource that is separate from the layout XML file Theme - A style applied to an entire Activity or app [in the app's manifest file]. When a style is applied as a theme, every view in the activity or app applies each style attribute that it supports automatically. Material Design - A comprehensive guide for visual, motion, and interaction design across platforms and devices. Important features are ToolBars, Floating Action Buttons, Snackbar, AppBarLayout, etc.
What is chain	Way to structure our View's is through chaining. We can chain multiple View's into a linked bi-directional position constraint. There are 3 main styles of chaining by specifying the attribute: Spread: views are evenly distributed Packed: views are packed together Spread inside: first and last view are affixed to the constraints on each end of the chain but rest are evenly distributed

FIT2081 chap 4,5,6

SMS receiver class, declarations and onReceive()	String[]{Manifest.permission.SEND_SMS, Manifest.permission.RECEIVE_SMS, Manifest.permission.READ_SMS}, 0); onReceive(): SmsMessage[] messages= Telephony.Sms.Intents.getMessagesFromIntent(intent);

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

FIT2081 chap 4,5,6

CoordinatorLayout, FrameLayout, GridLayout, LinearLayout, RelativeLayout and indirect subclasses such as TableLayout. Note: since a ViewGroup is a View, a ViewGroup can be contained by another ViewGroup.

Layout Parameters:
XML layout attributes defining layout parameters for the View that are appropriate for the ViewGroup in which it resides. Every ViewGroup class implements a nested class that extends ViewGroup.LayoutParams – this subclass contains properties that define the size and position for each child View, as appropriate for the view group.

ConstraintLayout:

solutions to nested layouts, views are all linearly placed

Components of layout/constraint layout

Views are attached to the layout through software analogues of springs or through horizontal and vertical guidelines

These springs can expand and collapse
Tension on springs holding a View between two endpoints can be biased towards

ConstraintLayout.LayoutParams
match_constraint – layout size is defined by the size of the constraint hence is always 0dp

wrap_content – size is defined by the required caption size of the view

FIT2081 chap 4,5,6

a)Returning true or false from onCreateOptionsMenu is need to let Android knows if the inflation of a menu layout from the method was handled successfully and not affecting other layouts

a) What is the role of return true or false from the callback method onCreateOptionsMenu?

b) If you have a navigation view with 3 items, how many events handlers do you need to code to handle the click events from these four items? And Why?

c) Suppose you are trying to link a button to its event handler. Briefly explain two differences between implementing the OnClickListener and using the android:onClick attribute in the activities XML file.

b) the number of events handlers depends on how each of 3 items listen to its respective events. If they are all listening to one event (the user press one button but the developers expect different methods to be called) then 1 handlers is need. But if each listen to different events then at least 3 handlers are needed.

c)Differences between onClickListener and Android:onClick

1)android:onClick is a function in the XML file holding onto the method onClick and the function it want to invoke. The function have to pass a View class as attribute so to use the method onClick

2)OnClickListener is an interface, or a class that has its own instance variables and methods that any class could implement. The onClickListener implements the action of click event separatedly from the View's component (i.e a button) that trigger it

FIT2081 chap 4,5,6

a) Assume you have a left to right drawer in you activity, write a java statement that closes it.
b) Briefly describe the required steps to change the text colour of a listview from black to red

b)
In order to change the color of the listView; first; we need to open the layout declared in the adapter class i.e(adapter =new ArrayAddapter<> (this,android.R.layout.simple_list_item_1 (which is a plain layout)
The next step is to create a new xml file in the layout folder and named accordingly. During the creation process, it is important to remove the root element of constraint layout which android set to be the default when new layout is created
The last step is copy paste the xml code into the new layout, change the reference for layout in the adapter class. To change the color (black to red), this can be done by setting android:textColor=@color/"black"

Assignment Project Exam Help

https://tutorcs.com

WeChat: cstutorcs

FIT2081 chap 4,5,6

Assume you have an activity with a listview (id=unitslist) in it. Develop a piece of code that adds the following items to the listview:
• Monash
• S1
• 2021

```java
ListView listView;
ArrayList<String>myData;
ArrayAdapter<String>adapter;
@Override
.protected void onCreate(Bundle savedInstanceState){
.
.
setContentView(R.layout_activity_main2);
myData=new ArrayList<>();
listView=findViewById(R.id.unitlist)
myData.add("Monash");
myData.add("S1");
myData.add("2021");
adapter=new ArrayAdapter(this, android.R.layout.simple_list_item_1,myData);
listView.setAdapter(adapter);
}
}
```