



Free Questions for Associate-Android-Developer by certsdeals

Shared by Travis on 29-01-2024

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

In Android 8.0, API level 26, some APIs regarding notification behaviors were moved from Notification to NotificationChannel. For example, what should we use instead of NotificationCompat.Builder.setPriority() for Android 8.0 and higher?

Options:

- A- NotificationChannel.setPriority()
- B- NotificationChannel.setImportance()
- C- NotificationCompat.Builder.setImportance()

Answer:

B

Explanation:

<https://developer.android.com/training/notify-user/build-notification>

Question 2

Question Type: MultipleChoice

What do you want from Room when you create a DAO method and annotate it with @Delete?

Example:

```
@Dao
```

```
public interface MyDao {
```

```
    @Delete
```

```
    public void deleteUsers(User... users);
```

```
}
```

Options:

- A-** Room generates an implementation that inserts all parameters into the database in a single transaction.
- B-** Room modifies a set of entities, given as parameters, in the database. It uses a query that matches against the primary key of each entity.
- C-** Room removes a set of entities, given as parameters, from the database. It uses the primary keys to find the entities to delete.

Answer:

C

Question 3

Question Type: MultipleChoice

What do you want from Room when you create a DAO method and annotate it with @Update?

Example:

@Dao

```
public interface MyDao {
```

```
    @Update
```

```
    public void updateUsers(User... users);
```

```
}
```

Options:

A- Room generates an implementation that inserts all parameters into the database in a single transaction.

B- Room modifies a set of entities, given as parameters, in the database. It uses a query that matches against the primary key of each entity.

C- Room removes a set of entities, given as parameters, from the database. It uses the primary keys to find the entities to delete.

Answer:

B

Question 4

Question Type: MultipleChoice

What happens when you create a DAO method and annotate it with @Insert?

Example:

@Dao

```
public interface MyDao {
```

```
    @Insert(onConflict = OnConflictStrategy.REPLACE)
```

```
    public void insertUsers(User... users);
```

```
}
```

Options:

- A-** Room generates an implementation that inserts all parameters into the database in a single transaction.
- B-** Room modifies a set of entities, given as parameters, in the database. It uses a query that matches against the primary key of each entity.
- C-** Room removes a set of entities, given as parameters, from the database. It uses the primary keys to find the entities to delete.

Answer:

A

Question 5

Question Type: MultipleChoice

What is demonstrated by the code below?

```
// RawDao.java
```

```
@Dao
```

```
interface RawDao {  
  
    @RawQuery  
    User getUserViaQuery(SupportSQLiteQuery query);  
  
}  
  
// Usage of RawDao  
  
...  
  
SimpleSQLiteQuery query =  
new SimpleSQLiteQuery("SELECT * FROM User WHERE id = ? LIMIT 1",  
new Object[]{userId});  
  
User user = rawDao.getUserViaQuery(query);  
  
...
```

Options:

A- A method in a Dao annotated class as a raw query method where you can pass the query as a SupportSQLiteQuery.

B- A method in a Dao annotated class as a query method.

C- A method in a RoomDatabase class as a query method.

Answer:

A

Question 6

Question Type: MultipleChoice

An example. In our ViewModelFactory (that implements ViewModelProvider.Factory) we have an instance of our Repository, named mRepository. Our ViewModel has such constructor:

```
public MyViewModel(MyRepository myRepository)...
```

Next, in our ViewModelFactory create ViewModel method (overriden) looks like this:

```
@NonNull
```

```
@Override
```

```
public T create(@NonNull Class modelClass) { try {
```

```
//MISSED RETURN VALUE HERE
```

```
} catch (InstantiationException | IllegalAccessException | NoSuchMethodException | InvocationTargetException e) {
```



```
throw new RuntimeException("Cannot create an instance of " + modelClass, e);  
}  
}
```

What should we write instead of "//MISSED RETURN VALUE HERE"?

Options:

- A-** return modelClass.getConstructor()
.newInstance(mRepository);
- B-** return modelClass.getConstructor(MyRepository.class)
.newInstance();
- C-** return modelClass.getConstructor(MyRepository.class)
.newInstance(mRepository);

Answer:

C

Question 7

Question Type: MultipleChoice

For example, we have a file in our assets folder `app/src/main/assets/sample_teas.json`. To get an `InputStream` for reading it, from our `Context` context, we can try doing this:

Options:

- A- `InputStream input = context.getResources().openRawResource(R.raw.sample_teas);`
- B- `InputStream input = context.getAssets().open('sample_teas.json');`
- C- `InputStream input = context.getResources().getAssets().open ('sample_teas.json');`

Answer:

B

To Get Premium Files for Associate-Android-Developer Visit

<https://www.p2pexams.com/products/associate-android-developer>

For More Free Questions Visit

<https://www.p2pexams.com/google/pdf/associate-android-developer>

