CAT1

Presentation

This Continuous Assessment Test (CAT) is an introduction to Android app development. In this activity, you will install Android Studio; become familiar with the development environment; and explore the basics of the Kotlin programming language.

Competencies

This CAT will develop the following competencies of the Bachelor's degree in Techniques for Software Development:

- Adapt to new software development technologies and to future environments, updating professional skills.

Objectives

The learning outcomes of this CAT are the following:

- Understand the mobile ecosystem and get to know the platforms, technologies and key players in the mobile industry.
- Install and use an IDE (integrated development environment) for mobile app development (Android Studio).
- Code using a programming language for developing native mobile apps (Kotlin).

Problem statement

1. **API levels** (Weight: 20%)

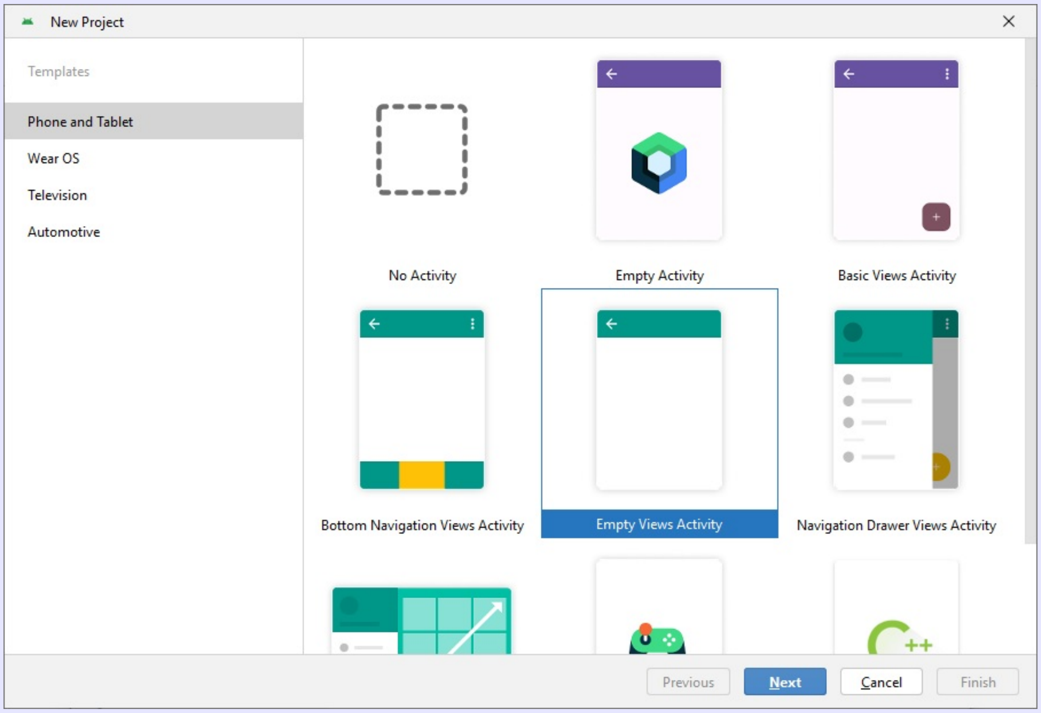
   Each Android API level offer different sets of features, with more recent API levels offering more capabilities. Explain from which API level it is possible to perform the following:

   (a) Animated vector graphics.

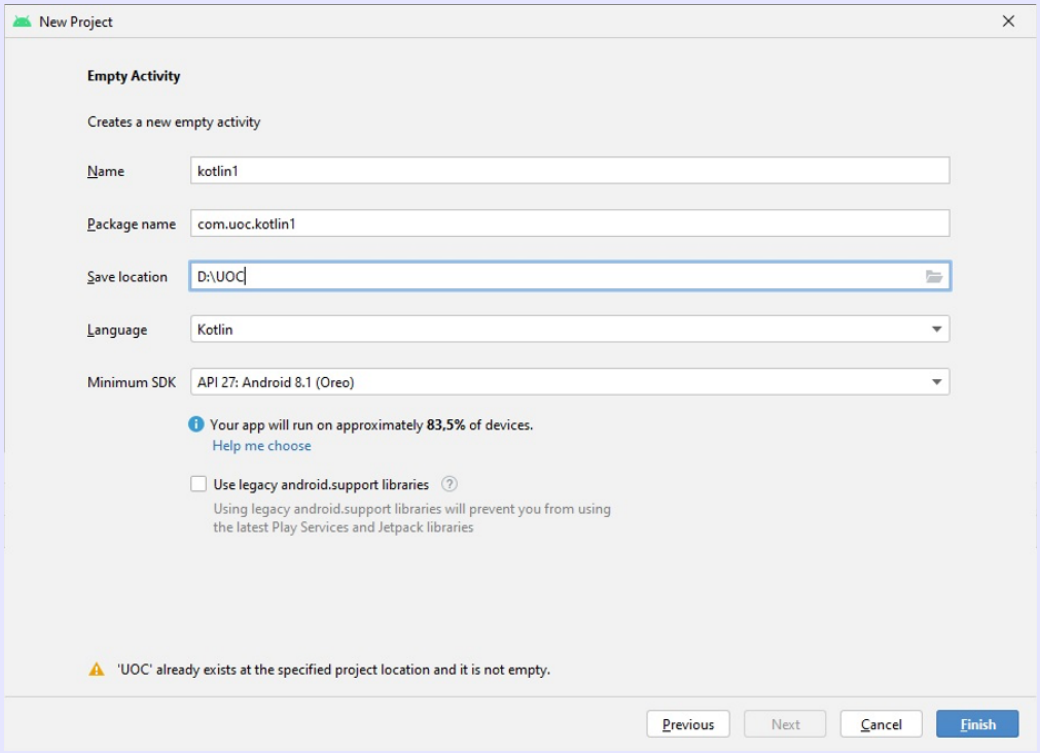
   (b) Full-screen applications.

In order to solve the remaining exercises of the CAT, we will need to create an Android app.

- First of all, we will need to install Android Studio and then create an app (using the menu option `File/New/New Project` or `New Project` from the starting dialog) of type `Empty Views Activity`.



- We can use the following settings for this project:



New Project

Empty Activity

Creates a new empty activity

Name: kotlin1

Package name: com.uoc.kotlin1

Save location: D:\UOC

Language: Kotlin

Minimum SDK: API 27: Android 8.1 (Oreo)

Your app will run on approximately **83,5%** of devices.
Help me choose

☐ Use legacy android.support libraries
Using legacy android.support libraries will prevent you from using the latest Play Services and Jetpack libraries

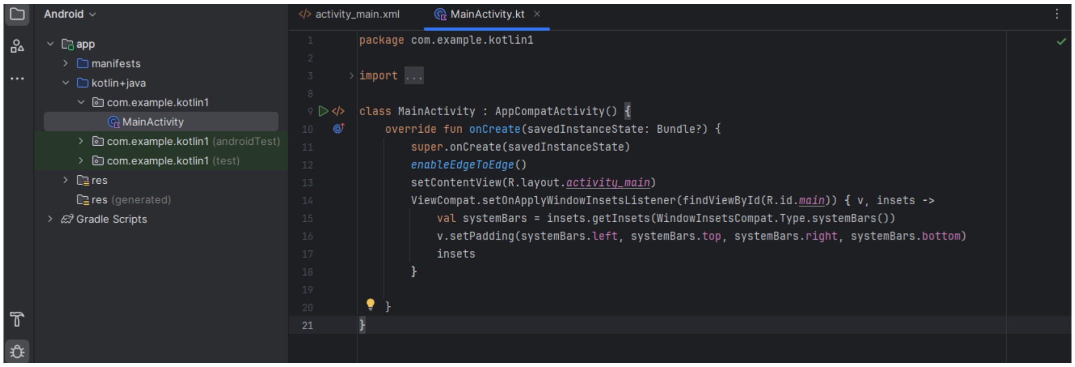
'UOC' already exists at the specified project location and it is not empty.

- Using the Device Manager, create a device of type Pixel 7, API level 33 and with Google API. Test the app you have created on this device, as you will be using it to solve the remaining exercises.

2. **Kotlin fundamentals** (Weight: 20%)

   In this exercise, we will look at simple errors in Kotlin code and how they can be fixed.

   Find the class `MainActivity` in the `Empty Views Activity` application we just created. There, look for method `override fun onCreate(savedInstanceState: Bundle?)`. Position yourself at the end of this method.



```kotlin
package com.example.kotlin1

import ...

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets
        }

    }
}
```

Then, copy each of the following (incorrect) code snippets one by one. For each one, explain why they do not work and propose a fix that solves the problem. You can leave the erroneous code commented, adding your explanation within the comment. You can only add the symbol '?' to the left side of the assignment statement if you need it. You must do the other modifications on the right side. In some cases, there may be more than one correct answer. You should choose the one that has the lowest CPU cost. The "as" cast is the lowest cost operator because it does nothing at runtime. The "as" operator only acts at compile time.

(a)
```kotlin
var v4:Int = 4
var v5:Double = v4;
```

(b)
```kotlin
var v2:Any = 4.4
var v3:Double = v2
```

(c)
```kotlin
var v6:Int = null
```

3. **Collections in Kotlin** (Weight: 20%)

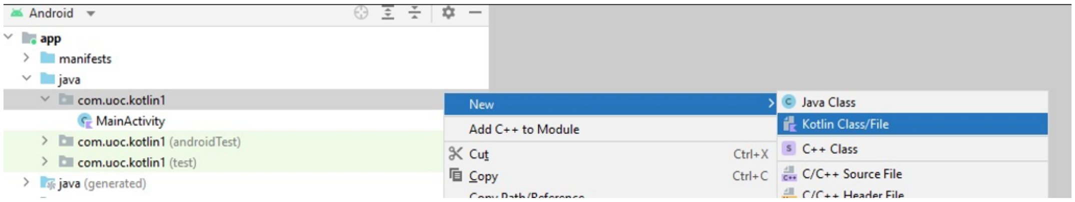
   In this exercise, we will solve simple tasks using Kotlin collections (again in the context of the `MainActivity`).

   (a) We want to access a car's price as quickly as possible using its model name. What data structure should we use?. Then, add a car with its price to the structure and query its price using the model name.

   (b) Create a list of String called `car_name_list`. Add 5 different items to the list and then remove the string in the second position. Finally, iterate through the list and print the value for each position in the `Logcat` window using `Log.d("debug",v)`.

4. **Classes in Kotlin** (Weight: 30%)

   In this exercise, we will create and extend Kotlin classes (in the context of the app we created).



   (a) Create a class called `Car`. It will include an attribute `name` of type string and an attribute `price` of type Int. Its constructor will have two parameters called `pname` and `pprice` that will initialize the previously mentioned attributes.

   (b) Create a class called `User`. It will include an integer attribute called `id` and a string attribute called `username`. Its constructor will have a parameter called `pid` that will initialize the previously mentioned `id` attribute and username as null. Moreover, this class will have an attribute called `cars` of type `HashMap<String,Car>`. The HashMap key is the Car name.

   (c) In class `User`, add a method with signature `fun addCar(d:Car)` that will allow adding its parameter `d` to the attribute of type `HashMap<String,Car>`.

   (d) In class `User`, add a method with signature `fun removeCar(d:String)` that will allow removing the Car with name `d` from the attribute of type HasMap `cars`.

   (e) Why has the price been declared as type Int in the class `Car` instead of type double? Hint: What units are we storing?

5. **Use classes** (Weight: 10%)

   Going back to the `MainActivity` of our app, perform the following tasks:

   (a) Create a User with id `18`.

   (b) Create a Car with name `"Ferrari Purosangue"` and price `39835000`.

   (c) Add the Car to the information about the user.

   (d) Remove the previous Car from the information about the user.

Resources

Basic Resources

- Course Wiki: Section 1 - Introduction
- Course Wiki: Section 2 - Android Studio
- Course Wiki: Section 3 - The Kotlin programming Language

Additional Resources

- Official Android developer documentation
- Official Android Studio documentation

Assessment criteria

- All exercises in this CAT must be solved **individually**.
- Each exercise has the following weight:
  1. 20%
  2. 20%
  3. 20%
  4. 30%
  5. 10%
- The use of artificial intelligence tools is not allowed in this activity.

  The course plan and the UOC's site on academic integrity and plagiarism have information on what is considered misconduct in assessment activities and the consequences this may have.

Submission format and deadline

You must submit a PDF file including the answers to written exercises. The name of the file you submit must be **CAT1_SurnameName.PDF** and you should include your full name within the PDF. This document must be submitted in the **"Submission of the assignment C1"** of the "Modules" section of the classroom. You must submit the assignment **before 23:59 of 2025/03/09**. **Late submissions will not be accepted.**