

Nom : ..... Prénom: ..... Classe: 4 IIRG.....

Matière :	Programmation Mobile	Filière : IIR Niveau : 4 <sup>e</sup>
Date de l'épreuve :	21/06/2025 14:00 - 16:00	Durée de l'épreuve : 2H00

**Consignes à Respecter**

- La clarté et la simplicité des réponses et du code seront prises en considération.
- Toute communication entre candidats est formellement interdite.
- Tout appareil électronique (smartphone, montre connectée, tablette, etc.) doit être éteint et rangé hors de portée.

**Partie I : Questions de Cours (6 pts).**

1. Expliquer le rôle de **enqueue** dans la bibliothèque **Retrofit** ?

---

---

---

---

2. Citer et décrire les permissions de la **Géolocalisation** dans une application Android.

---

---

---

---

3. Quelle est la différence entre les permissions déclarées dans le fichier Manifest et celles dans le code Java/Kotlin?

---

---

---

---

4. Quel est le rôle du dossier "**drawable**" ?

---

---

---

---

5. Quelle est l'utilité de comprendre le cycle de vie d'une **Activity** ?

---

---

---

---

6. Quelle est la différence entre définir l'interface graphique d'une Activity dans un fichier layout en XML et la définir dans le code Java ?

---

---

---

---

## Partie II : Analyse de code Android (6 pts)

Soit une Activity StudentsActivity.java

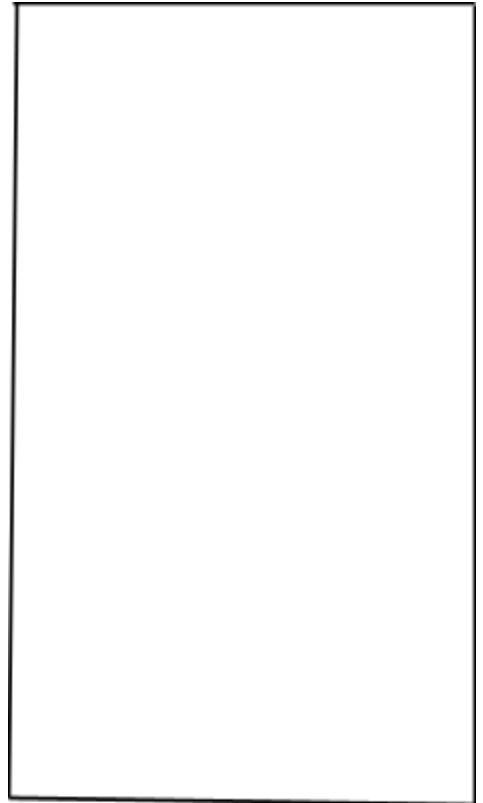
```
public class StudentsActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_students);  
        // Données de test (supposées venir d'une API)  
        ArrayList<Student> students = new ArrayList<>();  
        students.add(new Student( prenom: "Abbas", nom: "IBN FARNAS"));  
        students.add(new Student( prenom: "Abir", nom: "NAJAH AL AMIR"));  
        students.add(new Student( prenom: "Safae", nom: "SIDARJINE"));  
        LinearLayout studentsLayout = findViewById(R.id.studentsLayout);  
        for (Student student : students) {  
            TextView textView = new TextView( context: this);  
            textView.setText(student.getPrenom() + " - " + student.getNom());  
            textView.setPadding( left: 16, top: 8, right: 16, bottom: 8);  
            textView.setTextSize(16);  
            studentsLayout.addView(textView);  
        }  
    }  
}
```

et son fichier de layout activity\_students.xml :

```
<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:padding="16dp">  
  
    <LinearLayout  
        android:id="@+id/studentsLayout"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:orientation="vertical" />  
</ScrollView>
```

PS. Un ScrollView est un conteneur (invisible) qui permet de scroller son contenu.

1. Dans le cadre ci-contre, dessinez grossièrement la structure de cette Activity une fois lancée. **(1,5pt)**
2. Quelle est la différence entre la solution proposée et l'utilisation d'une ListView à la place? Précisez laquelle des deux solutions est la meilleure et justifiez votre réponse. **(1pt)**



3. Nous souhaitons remplacer notre solution par une **ListView** (ou RecyclerView). Expliquez toutes les étapes/mécanismes nécessaires pour faire cette modification (pas besoin de donner le code). **(2pt)**

4. Dans notre layout XML, nous avons précisé le "padding" en "dp". **(1,5pt)**

a. Que signifie "dp" ?

b. Pourquoi cette unité de mesure (dp) est préférable au "px" dans ce cas d'utilisation?

c. Quelle est la différence entre "dp" et "sp"?

### Partie III : Production de code Android (8 pts)

### Exercice 1: (Persistance des données)

On suppose qu'on a déjà une table Client dans une base de données SQLite avec les champs suivants : Client(id, nom, email, age). Le champ id est auto-incrémenté.

L'insertion doit être faite **sans utiliser de requête SQL directe**, mais en utilisant la méthode générique **insert(...)** fournie par SQLiteDatabase.

- Implémentez la fonction **insertClient(...)** qui permet d'exécuter l'instruction suivante : **(3pt)**

```
INSERT INTO Client (nom, email, age) VALUES ('Ahmed', 'ahmed@example.com', 30);
```

[illegible]

## Exercice 2: (Évènements et Intents)

On souhaite créer une petite application contenant :

- un champ de saisie pour entrer un numéro de téléphone,
- un bouton permettant d'ouvrir l'application téléphone pour composer automatiquement ce numéro (mais sans l'appeler directement).

1. Complétez le code pour afficher un Toast quand on effectue un clic sur le bouton "Appeler", avec le message suivant : "Tentative d'ouverture de l'application Téléphone..." (2pt)

```
Button btnAp = .....(R.id.btnAp1);
```

```
btnAp .....(v -> {  
    Toast.makeText(....., " Tentative d'ouverture de l'application  
Téléphone...", ..... ) .....();  
    return true;  
});
```

2. Qu'est-ce qu'un **Intent** en Android ? (1pt)

3. Complétez le code Java suivant pour créer un **Intent implicite** qui ouvre l'application Téléphone avec le numéro saisi dans un champ EditText ayant pour ID "phoneNumber". (2pt)

```
EditText edtPhone = findViewById(R.id.phoneNumber);
```

```
startActivity(intent);
```

## FIN DE L'EXAMEN

*En cas de besoin d'espace supplémentaire, mentionnez le numéro de la question, et complétez votre réponse sur cette page.*

[illegible]