

# Centres étrangers – 2026 – sujet2 - Correction

## Exercice 1 (6 points)

---

### Partie A

#### 1. Importer les variables et les classes

```
from donnees import vehicules, energie
from chauffeur import Chauffeur
from taxi import Taxi
from client import Client
from course import Course
```

👉 *Commentaire : Les éléments définis dans d'autres scripts doivent être importés avant de pouvoir être utilisés dans application.py.*

---

#### 2. Instancier le chauffeur John Doe

```
john = Chauffeur("Doe", "John", "140159320012")
```

👉 *Commentaire : Les paramètres doivent être fournis dans l'ordre défini par le constructeur : nom, prénom puis numéro NEPH.*

---

#### 3. Créer le véhicule et l'attribuer à John Doe

```
taxi_john = Taxi(
    "HG-818-AV",
    "standard",
    "électrique",
    False
)
```

```
taxi_john.choix_chauffeur(john)
```

👉 *Commentaire : Le véhicule est créé puis associé au chauffeur grâce à la méthode choix\_chauffeur.*

---

#### 4. Assertion dans le constructeur de `Course`

```
assert taxi.est_libre(), "Le taxi n'est pas disponible."
```

👉 *Commentaire : Cette assertion empêche la création d'une course avec un véhicule déjà occupé.*

---

#### 5. Marquer le taxi comme occupé

Dans le constructeur de la classe `Course` : `taxi.modifier_libre(False)`

👉 *Commentaire : Dès qu'une course est créée, le taxi ne doit plus être disponible pour une autre réservation.*

---

#### 6. Créer la course de Jeanne Doe

```
jeanne = Client(  
    "Doe",  
    "Jeanne",  
    "6 rue des ordinateurs, Paris",  
    2  
)
```

```
course_jeanne = Course(  
    jeanne,  
    taxi_john,  
    "6 rue des ordinateurs, Paris",  
    "211 avenue Jean Jaurès, Paris"  
)
```

👉 *Commentaire : L'objet client doit être créé avant l'objet course qui l'utilise.*

---

#### 7. Écrire la méthode `temps`

```
def temps(self):  
    return (self.date_arrivee - self.date_depart).total_seconds() / 60
```

👉 *Commentaire : La différence entre deux objets `datetime` est exprimée en secondes ; il faut donc diviser par 60 pour obtenir des minutes.*

---

## 8. Compléter la méthode `tarif`

```
def tarif(self):
    type_vehicule = self.taxi.type_vehicule
    donnees_tarifaire = vehicules[type_vehicule]

    prise_en_charge = donnees_tarifaire['prise_en_charge']
    tarif_h = donnees_tarifaire['tarif_horaire']
    prix_km = donnees_tarifaire['prix_km']

    tarif = prise_en_charge
    tarif = tarif + prix_km * self.distance()
    tarif = tarif + tarif_h * self.temps() / 60

    return tarif
```

👉 *Commentaire : Le coût total est la somme de la prise en charge, du coût kilométrique et du coût horaire.*

---