

Exercice 1: QCM (10 points)

1.1. Lequel des instructions suivantes sont invalide :

- (a) `_x = 1`
- (b) `__x = 1`
- (c) `__str__=1`
- (d) Toutes les réponses sont vraies

1.2. Une méthode privée est _____

- (a) Accessible de n'importe quelle classe
- (b) Accessible seulement à partir de la même classe

1.3. Quelle est la sortie du code suivant :

```
class Library:
    def __init__(self, name='lib') :
        self.name = name
    def practice(self, language='Java'):
        print(language)
l = Library()
l.practice('Python')
```

- (a) Python
- (b) Java
- (c) Language
- (d) Python Java

Programmation Orientée Objet

1.4. Que fait la fonction `__init__()` en Python ?

- (a) Initialise la classe pour l'utilisation.
- (b) Cette fonction est appelée lorsqu'un nouvel objet est instancié.
- (c) Initialise tous les attributs de données à zéro lors de l'appel.
- (d) Aucune de ces réponses n'est vraie.

1.5. Quelle est la sortie du code suivant :

```
class Point:

    def __init__(self, x = 0, y = 0):
        self.x = x
        self.y = y

    def __sub__(self, other):
        x = self.x + other.x
        y = self.y + other.y
        return Point(x,y)

point1 = Point(30, 40)
point2 = Point(10, 20)
point3 = point1 - point2
print(point3.x, point3.y)
```

- (c) 20 30
- (d) 40 60
- (e) 10 20
- (f) 20 40

1.6. L'encapsulation permet _____

- (a) D'avoir un contrôle complet sur ces données et sur des contraintes à imposer sur ces données
- (b) De documenter le code
- (c) De contrôler l'accès aux données du constructeur

1.7. Un accesseur est une méthode permettant de _____

- (a) Modifier le contenu d'un attribut privé
- (b) Récupérer le contenu d'un attribut privé

1.8. Quel est le rapport entre une classe et un objet ?

- (a) Un objet est une instance de classe
- (b) Un objet est une intendance de classe
- (c) Il n'ya aucun rapport

Programmation Orientée Objet

1.9. Le constructeur _____

- (a) Permet d'initialiser les valeurs des attributs d'une classe
- (b) Ne sert à rien
- (c) Permet de documenter la classe

1.10. Une méthode publique est :

- (a) Accessible de n'importe quelle classe
- (b) Accessible seulement à partir de la même classe

Exercice 2 : (3 points)

Ecrire une classe nommée « **Personne** » ayant trois attributs définissant certaines caractéristiques d'une personne réelle :

- Taille
- Poids
- age

En plus du constructeur, cette classe aura comme méthode :

- « *imc(self)* » qui détermine l'IMC de la personne,
- « *interpretation(self)* » qui affiche "Insuffisance pondérale" si l'IMC est inférieur ou égale à 18,5 et qui affiche "obésité" si l'IMC est supérieur ou égale à 30.

Rappel : l'IMC (Indice de masse corporelle) est donné par la formule $imc = \frac{poids}{taille^2}$ avec le poids en kg et la taille en m.

Implémenter cette classe en Python.

Programmation Orientée Objet

Exercice 3 : (7 points)

La formule de Minac et Willans permet de déterminer les nombres premiers. Elle définit par :

$$p_n = 2 + \sum_{m=2}^{2^n} \left[\frac{n}{1 + \sum_{k=2}^m \left[\frac{(k-1)!+1}{k} - \left[\frac{(k-1)!}{k} \right] \right]} \right]^{\frac{1}{n}}.$$

On propose d'implémenter une classe permettant de générer les nombres premiers grâce à cette formule. Ecrire la classe « MinacWillans » ayant les attributs suivants :

- N : un entier correspondant aux nombres premiers à générer

Implémenter les méthodes suivantes :

- « **factoriel** » prenant en paramètre un entier « m » puis retournant m !
- « **minac** » prenant en paramètre un entier « n » puis effectuant l'opération suivante :

$$\sum_{k=2}^m \left[\frac{(k-1)!+1}{k} - \left[\frac{(k-1)!}{k} \right] \right]$$

Nb : cette formule correspond à une partie du dénominateur de l'équation « p_n ». La méthode « minac » est **privée**

- « **generer** », ne prenant pas de paramètres mais retournant le résultat de la formule de Minac et Wilans.