

## Corrigé type :

### Exercice 1 (11,5 points) :

1. Choisissez la (ou les) réponse(s) correctes:

**1.1. Le type de retour du constructeur est: réponse juste 1p fausse 0**

d)Aucun type

**1.2. Java est un langage: une réponse juste 0,5, deux et trois réponses juste 0,75, tous juste 1p**

a) Interprété

b) Compilé

c) Multiplateforme

d) Sécurisé

**1.3. Par convention les constantes sont: une réponse juste 0,75 , tous juste 1 p**

a) Déclaré avec le mot clé final

b) Déclaré en majuscule

**1.4. Le casting implicite une réponse juste 0,5 deux justes 0,75 tous juste 1 p**

a) Se fait du plus petit type au plus grand type

c)Permet de changer le type d'une variable

d)Se fait automatiquement

**1.5. Le casting explicite réponse juste 1p fausse 0**

a) Se fait du plus grand type au plus petit type

**1.6. Une variable de classe est une variable :une réponse juste 0,5 deux justes 0,75 tous juste 1 p**

a) Partagée par toutes les instances de la classe

b) Précédée par le mot clé static

c) N'appartient pas à une instance particulière, elle appartient à une classe

**1.7. 1.7. Une méthode d'instance est une méthode qui :une réponse juste 0,5 deux justes ou trois 0,75 tous juste 1 p**

**1 p**

a) Agit sur les variables d'instance et ces propre variables

c) Appelée après avoir instancié un objet

d) N'est précédée par aucun mot-clé

**1.8. 1.8. Une classe abstraite est une classe qui: réponse juste 0,5 p fausse 0**

(c) Ne peut être instanciée.

**2.1**

(a) T1 == T2? (true) **1 p**

(b) T1 == T3?(false) **1p**

(c) T2 == T3?(false) **1p**

2.2. erreur de compilation **0,5 p**. justification: variable d'instance ne peut pas être appelée sans l'objet. **0,5 p**

**Exercice 2 (08,5):**

**1. 1 p**

```
class Salarié {  
    private String nom;  
    private String prénom;  
    private int age;  
    private double salaire;  
    private String grade ou int ou char;  
    private double prime;  
    private int nombreEnfants;
```

**2. 1 p**

```
    public Salarié(String nom, String prénom, int age, double salaire, String grade, double  
prime, int nombreEnfants) {  
        this.nom = nom;  
        this.prénom = prénom;  
        this.age = age;  
        this.salaire = salaire;  
        this.grade = grade;  
        this.prime = prime;  
        this.nombreEnfants = nombreEnfants;  
    }  
}
```

**3.**

**3.1. 1 p**

```
    public double AF() {  
        return nombreEnfants * 300;  
    }  
}
```

**3.2. 1 p**

```
    public void afficher() {  
        System.out.println("Nom: " + nom);  
        System.out.println("Prénom: " + prénom);  
        System.out.println("Age: " + age);  
        System.out.println("Salaire: " + salaire);  
        System.out.println("Prime: " + prime);  
    }  
}
```

3.3. **0,5 p**

```
public void calculerPrime() {  
    if (salaire > 5000) {  
        prime = 0.1 * salaire;  
    } else {  
        prime = 0.05 * salaire;  
    }  
}
```

3.4. **0,5 p**

```
public double calculerPrimeSpeciale(boolean performance, int ancienneté) {  
    if (performance && ancienneté >= 5) {  
        return 0.15 * salaire;  
    } else if (performance || ancienneté >= 5) {  
        return 0.1 * salaire;  
    } else {  
        return 0;  
    }  
}
```

3.5. **1 p**

```
public void augmenterSalaire(double pourcentage) {  
    salaire += (pourcentage / 100) * salaire;  
}
```

3.6. **0,5 p**

```
public double calculerSalaireNet(double tauxImpot) {  
    double impots = ((tauxImpot/100) * salaire);  
    return salaire - impots;  
}
```

3.7. **1 p**

```
public String getNom() {  
    return nom;  
}  
  
public double getPrime() {  
    return prime;  
}  
  
public double getSalaire() {  
    return salaire;  
}
```

3.8. **1 p**

```
public boolean mêmePrime(Salarié autreSalarié) {  
    If (prime= autreSalarié.calculerprime())  
return true; else return false;  
}
```

```
}
```

### **BONUS 1 p**

```
public class TestSalarié {  
    public static void main(String[] args) {  
        Salarié salarié1 = new Salarié("ali", "bahi", 35, 6000, "Manager", 0, 2);  
        Salarié salarié2 = new Salarié("Mohamed", "said", 28, 4000, "Assistant", 0, 0);  
  
        salarié1.afficher();  
        System.out.println("Allocation Familiale: " + salarié1.AF());  
  
        salarié1.calculerPrime();  
        System.out.println("Prime calculée: " + salarié1.getPrime());  
  
        double primeSpeciale = salarié1.calculerPrimeSpeciale(true, 7);  
        System.out.println("Prime spéciale calculée: " + primeSpeciale);  
  
        double salaireNet = salarié1.calculerSalaireNet(0.2);  
        System.out.println("Salaire net: " + salaireNet);  
        System.out.println("Les deux salariés ont la même prime: " + salarié1.mêmePrime(salarié2)); }  
}
```