

Prénom et Nom : _____

1 Questions de cours (10 points)

1. 2 points Quels sont les 2 principaux niveaux de visibilité disponibles en java ?



2. 1 point Qu'est-ce que la surcharge ?



3. 2 points Soit A et B extends A deux classes. Sélectionnez les instructions valides.

— A a = new B();

— B b = new A();



4. 1 point Comment s'appelle la possibilité offerte dans la question précédente ?



5. 1 point Quelle est la différence entre une interface et une classe abstraite ?



6. 1 point Une classe peut-elle avoir plusieurs classes mères ? Existe-t-il des cas particuliers ?



7. 2 points Étant donnée l'interface suivante :

```
public interface MonInterface
{
    public void f();
}
```

Quels sont les trois moyens permettant au code suivant de passer la compilation ?

```
public class MaClasse implements MonInterface
{
}
```



2 Températures (10 points)

Les classes suivantes servent à modéliser des températures exprimées dans plusieurs unités. Nous rappelons qu'étant donnée une température exprimée en degrés Celsius, on obtient la même en degrés Kelvin en additionnant 273, et on la convertit en degrés Fahrenheit en multipliant par 1.8 puis en additionnant 32.

1. 10 points Complétez le code source suivant :

```
interface Comparable
{
    /*
     * Retourne un nombre negatif si this < other,
     * positif si this > other, 0 si this = other.
     */
    public int compareTo(Comparable other);
}

..... class Temperature ..... Comparable
{ // 1 point
    private double value;

    public double getValue()
    {
        .....
    } // 1 point

    public String toString()
    {
        return " " + value;
    }

    public Temperature(double value){
        this.value = value;
    }

    public abstract double enCelsius();

    public int compareTo(Comparable other)
    {
        double t = enCelsius(),
            o = .....
        // 1 point

        return .....
    } // 1 point
}

class Celsius extends Temperature
{
    public Celsius(double value){
        super(value);
    }

    public Celsius(Temperature t){
        this(t.enCelsius());
    }


    public double enCelsius(){
        return getValue();
    }

    public String toString(){
        return super.toString() + " degrés Celsius";
    }
}

class Kelvin extends Temperature
{
    public Kelvin(double value){
        super(value);
    }
}
```


```


    public Kelvin(Temperature t)
    {
        this(t.enCelsius() + 273);
    }


    public double enCelsius()
    {
        return getValue() .....  .....
    } // 1 point

    public String toString()
    {
        return super.toString() + " degrés Kelvin";
    }
}

class Farenheit extends Temperature
{
    public Farenheit(double value)
    {
        super(value);
    }

    public Farenheit(Temperature t)
    {
        .....  .....
    } // 2 points

    public double enCelsius()
    {
        .....  .....
    } // 2 points

    public String toString()
    {
        .....  .....
    } // 1 point
}

public class Test
{
    public static void main(String[] args)
    {
        Temperature celsius = new Celsius(37),
        kelvin = new Kelvin(celsius),
        farenheit = new Farenheit(kelvin),
        check = new Celsius(farenheit);
        System.out.println(celsius + "\n" + kelvin + "\n"
            + farenheit + "\n" + check);
    }
}

```