

Programmation I

Test de mi-parcours, Version: A

Nom: \_\_\_\_\_

Carte d’étudiant: \_\_\_\_\_

*Pour chaque question encerclez la bonne réponse; si le temps permet, remplissez la table*

Question générale	1	2	3	4	5	6	Question spécifique	1	2	3	4	5
Réponse							Réponse					

Section 1. Questions générales bonne réponse=1pt ; mauvaise réponse=-0.5pt ; pas de réponse=0pt.

*Bonne réponse=1pt; mauvaise réponse=-0.5pt; pas de réponse=0pt.*

1. Lequel de ces langages n’est pas orienté objet

- (a) Java
- (b) Simula
- (c) C
- (d) C++

2. class B {

```
private int x; private float y;  
public void setX( int x) {this.x=x; }  
public void setY( float y) {this.y=y; }  
}
```

Le code ci-dessus est une illustration

- (a) de l’encapsulation
  - (b) de la cryptographie
  - (c) du polymorphisme
  - (d) de l’héritage
3. Toutes les classes de Java héritent de
- (a) la classe Base
  - (b) l’interface Cloneable
  - (c) la classe Object
  - (d) la classe Class
4. Le passage de paramètres en Java est effectué par
- (a) variable
  - (b) référence
  - (c) valeur
  - (d) nom
5. Un objet traite un message reçu
- (a) en implémentant une interface
  - (b) en exécutant un constructeur
  - (c) en exécutant une méthode statique
  - (d) en exécutant une méthode d’instance
6. La fonction principale de la JVM consiste à
- (a) compiler le code Java
  - (b) interpréter le bytecode
  - (c) virtualiser les appels système
  - (d) fournir la console Java

Section 2. Questions spécifiques bonne réponse=3pt; mauvaise réponse=-1pt; pas de réponse=0pt

1. Pour les classes Oeuf et Poule définies comme suit :

```
class Oeuf {
    public int x;
    public Oeuf() {x=5; }
    public Oeuf(int y) {x=y; }
}

class Poule extends Oeuf {
    public Poule() {}
    public Poule(int i){this(); x=x*i; }
    public Poule(String s){super(33); x- -; }
}
```

qu'affichera le code suivant ?

```
Poule b1=new Poule("2004"); Poule b2 =new Poule(2004); Poule b3= new Poule();
System.out.println(b1.x + " et " + b2.x + " et encore " + b3.x );
```

- (a) 32 et 10020 et encore 5
  - (b) 32 et 0 et encore 0
  - (c) -1 et 2004 et encore 5
  - (d) autre chose (laquelle?)
2. Pour la classe D définie comme suit :

```
class D {
    public static int x;
    public int y;
    public static travailler() {x++;}
    public D() {x++; y- -; }
}
```

qu'affichera le code suivant ?

```
D.travailler(); D a=new D(); D b=new D(); a.travailler();
System.out.println(b.x + " et " + b.y);
```

- (a) 2 et -1
  - (b) ce code ne se compile pas
  - (c) 4 et -2
  - (d) 4 et -1
3. Combien d'instances de la classe A sont créées pendant l'exécution du code suivant ? Combien en reste après le passage du *Garbage collector* ?

```
A u,b,c;
A a=new A();
b=new A(); c=b;
a=b;
```

- (a) 5;2
  - (b) 2;2
  - (c) 2;1
  - (d) 3;3
4. Pour les classes A et B définies comme suit :

```
class A {
    public int f(int x) {return(x+1) };
    public static int g(int x) {return (6); }
}

class B extends A {
    public int f(int x) {return(x+2) };
    public static int g( int x) {return (x+4); }
}
```

qu'affichera le code suivant ?

```
B b=new B(); A a =b;
System.out.println(a.f(2)*a.g(3));
```

- (a) 18
  - (b) 21
  - (c) 24
  - (d) 28
5. Étant donné que la classe Sardine étend la classe Poisson, trouvez une ligne qui passe bien la compilation mais produit une erreur à l'exécution parmi les suivantes
- (a) Poisson y =new Poisson(); Sardine x= (Sardine)y; Poisson z=x;
  - (b) Sardine y =new Sardine(); Poisson x= y; Sardine z=(Sardine)x;
  - (c) Poisson y =new Sardine(); Object x= y; Sardine z=x;
  - (d) Poisson y =new Poisson(); Sardine z= new Sardine(); y=z;