

POO 2-L2 MI LI  
QCM 1, Version: A

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

Carte d’étudiant: \_\_\_\_\_

Cursus: \_\_\_\_\_

*Remplissez la table avec les lettres correspondant à vos réponses.*

Questions	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Réponse(s)																						

*Bonne réponse=1pt; mauvaise réponse ou réponse incomplète =-0,5pt; pas de réponse=0pt.*

- On considère la classe définie par : `class A {int i; void print(){System.out.println(i); }}`  
L’exécution de `A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();`
  - affiche 5
  - affiche 10
  - la classe A ne peut pas être compilée
- Avec la méthode: “ `static Integer f(int i){Integer n=i; return n;}` ” le morceau de code:  
“`Integer n1=2; if (n1==f(n1))System.out.println("égal"); else System.out.println("non égal");` ”
  - affichera égal
  - affichera non égal
  - provoquera une erreur à la compilation ou à l’exécution
- Avec la classe: “`class B{int i;B(int i){this.i=i;}}` ” le morceau de code:  
`B a=new B(3); B b=new B(0); b.i=a.i;`  
`if(a==b) System.out.println("égal"); else System.out.println("non égal");`
  - affichera égal
  - affichera non égal
  - provoquera une erreur à la compilation ou à l’exécution

4. En java:
  - (a) toute instruction a une valeur
  - (b) toute expression a un type
5. Le “byte code” java:
  - (a) est indépendant de la plateforme
  - (b) est un fichier binaire exécutable
6. Les déclarations:
 

```
static void f1(int i) {System.out.println("void f1(int)");}
static int f1(int i) {System.out.println("int f1(int)");}
```

 et le morceau de code: “int i= f1(new Integer(2));”
  - (a) afficheront int f1(int)
  - (b) afficheront void f1(int)
  - (c) provoqueront une erreur à la compilation ou à l’exécution
7. En supposant que **a** et **b** sont des variables correctement déclarées “**a=b+1**” est:
  - (a) une expression
  - (b) une instruction
8. On considère la classe définie par : `class A {static int i; static void print(){System.out.println(i); }}`.  
 L’exécution de:  
`A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();`
  - (a) affiche 5
  - (b) affiche 10
  - (c) la classe **A** ne peut pas être compilée
9. Avec la méthode: “`static void perm(Integer i,Integer j){Integer tmp; tmp=j; j=i; i=tmp;}`” le morceau de code:  
`int i1=10,i2=5; perm(i1,i2); System.out.println(i1);`
  - (a) affichera 10
  - (b) affichera 5
  - (c) provoquera une erreur à la compilation ou à l’exécution
10. Les déclarations:
 

```
static void f1(int i) {System.out.println("void f1(int)");}
static void f1(Integer i) {System.out.println("void f1(Integer)");}
```

 et le morceau de code: “f1(2)”
  - (a) afficheront void f1(int)
  - (b) afficheront void f1(Integer)
  - (c) provoqueront une erreur à la compilation ou à l’exécution
11. Le code:
 

```
Integer m=new Integer(2); Integer n=m;
if (m==n) System.out.println("égal");else System.out.println("non égal");
```

  - (a) affichera égal
  - (b) affichera non égal

12. Pour la classe D définie comme suit:

```
classD {  
    public int x;  
    public D() {x=3; };  
    public D( int a){this(); x=x+a;};  
    public D( int a, int b){this(b); x= x-a;}  
}
```

qu'affichera le code suivant?

```
D a=new D(5,6);  
System.out.println(a.x);
```

- (a) 1
- (b) 2
- (c) 3
- (d) 4

13. Le code:

```
Integer m=new Integer(2); Integer n=new Integer(2);  
if (n==m) System.out.println("égal"); else System.out.println("non égal");
```

- (a) affichera égal
- (b) affichera non égal

14. Quel est le résultat du morceau de code : `int i,j; i=10; j=i; j=5; System.out.println(i);`

- (a) il affiche 5
- (b) il affiche 10
- (c) il provoque une erreur

15. En java:

- (a) toute instruction a un type
- (b) toute expression a une valeur

16. Avec la classe: “`class B{int i;B(int i){this.i=i;}}`” et la méthode:

“`static void permutebis(B a,B b){int tmp=a.i;a.i=b.i;b.i=tmp; }`”

le morceau de code:

```
B a=new B(3); B b=new B(0); permutebis(a,b);System.out.println(a.i);
```

- (a) affichera 3
- (b) affichera 0
- (c) provoquera une erreur à la compilation ou à l'exécution

17. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=a;((A)o).i=10;;`

- (a) est correct (il peut être compilé et exécuté)
- (b) n'est pas correct (il ne peut pas être compilé et exécuté)

18. En java:

- (a) le type d'une expression est toujours déterminé à la compilation
- (b) le type d'une expression peut changer au cours de l'exécution

19. Avec la classe: `“class B{int i;B(int i){this.i=i;}}”` le morceau de code:  
`B a=new B(3); B b=a;`  
`if(a==b) System.out.println("égal"); else System.out.println("non égal");`
- (a) affichera égal
  - (b) affichera non égal
  - (c) provoquera une erreur à la compilation ou à l'exécution
20. La commande `javac` appliquée à un programme source syntaxiquement correct:
- (a) génère un fichier exécutable sur la machine cible
  - (b) génère un fichier qui peut être interprété par la machine virtuelle java
21. En général, une *applet* java:
- (a) est un code java destiné à être exécuté sur un navigateur web
  - (b) est un code java destiné à être exécuté sur un serveur web
22. On considère la classe définie par : `class A {static int i; }`. L'exécution de:  
`A a=new A(); A b=new A(); a.i=10; b.i=5; System.out.println(a.i);`
- (a) affiche 5
  - (b) affiche 10
  - (c) provoque une erreur

# Answer Key for Exam A

*Bonne réponse=1pt; mauvaise réponse ou réponse incomplète =-0,5pt; pas de réponse=0pt.*

1. On considère la classe définie par : `class A {int i; void print(){System.out.println(i); }}`  
L'exécution de `A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();`  

☐ (a) affiche 5

☐ (b) affiche 10

☐ (c) la classe A ne peut pas être compilée
2. Avec la méthode: “ `static Integer f(int i){Integer n=i; return n;}`” le morceau de code:  
“`Integer n1=2; if (n1==f(n1))System.out.println("égal"); else System.out.println("non égal");`”  

☐ (a) affichera égal

☐ (b) affichera non égal

☐ (c) provoquera une erreur à la compilation ou à l'exécution
3. Avec la classe: “`class B{int i;B(int i){this.i=i;}}`” le morceau de code:  
`B a=new B(3); B b=new B(0); b.i=a.i;`  
`if(a==b) System.out.println("égal"); else System.out.println("non égal");`  

☐ (a) affichera égal

☐ (b) affichera non égal

☐ (c) provoquera une erreur à la compilation ou à l'exécution
4. En java:  

☐ (a) toute instruction a une valeur

☐ (b) toute expression a un type
5. Le “byte code” java:  

☐ (a) est indépendant de la plateforme

☐ (b) est un fichier binaire exécutable
6. Les déclarations:  
`static void f1(int i) {System.out.println("void f1(int)");}`  
`static int f1(int i) {System.out.println("int f1(int)");}`  
et le morceau de code: “`int i= f1(new Integer(2));`”  

☐ (a) afficheront int f1(int)

☐ (b) afficheront void f1(int)

☐ (c) provoqueront une erreur à la compilation ou à l'exécution
7. En supposant que `a` et `b` sont des variables correctement déclarées “`a=b+1`” est:  

☐ (a) une expression

☐ (b) une instruction
8. On considère la classe définie par : `class A {static int i; static void print(){System.out.println(i); }}`.  
L'exécution de:  
`A a=new A(); A b=new A(); a.i=5; b.i=10; a.print();`  

☐ (a) affiche 5

☐ (b) affiche 10

☐ (c) la classe A ne peut pas être compilée

9. Avec la méthode: “static void perm(Integer i,Integer j){Integer tmp; tmp=j; j=i; i=tmp;}” le morceau de code:

```
int i1=10,i2=5; perm(i1,i2); System.out.println(i1);
```

- ☐ (a) affichera 10
- ☐ (b) affichera 5
- ☐ (c) provoquera une erreur à la compilation ou à l'exécution

10. Les déclarations:

```
static void f1(int i) {System.out.println("void f1(int)");}  
static void f1(Integer i) {System.out.println("void f1(Integer)");}  
et le morceau de code: “f1(2)”
```

- ☐ (a) afficheront void f1(int)
- ☐ (b) afficheront void f1(Integer)
- ☐ (c) provoqueront une erreur à la compilation ou à l'exécution

11. Le code:

```
Integer m=new Integer(2); Integer n=m;  
if (m==n) System.out.println("égal");else System.out.println("non égal");
```

- ☐ (a) affichera égal
- ☐ (b) affichera non égal

12. Pour la classe D définie comme suit:

```
classD {  
    public int x;  
    public D() {x=3; };  
    public D( int a){this(); x=x+a;};  
    public D( int a, int b){this(b); x= x-a;}  
}
```

qu'affichera le code suivant?

```
D a=new D(5,6);  
System.out.println(a.x);
```

- ☐ (a) 1
- ☐ (b) 2
- ☐ (c) 3
- ☐ (d) 4

13. Le code:

```
Integer m=new Integer(2); Integer n=new Integer(2);  
if (n==m) System.out.println("égal"); else System.out.println("non égal");
```

- ☐ (a) affichera égal
- ☐ (b) affichera non égal

14. Quel est le résultat du morceau de code : int i,j; i=10; j=i; j=5; System.out.println(i);

- ☐ (a) il affiche 5
- ☐ (b) il affiche 10
- ☐ (c) il provoque une erreur

15. En java:
- (a) toute instruction a un type
  - ☐ (b) toute expression a une valeur
16. Avec la classe: `class B{int i;B(int i){this.i=i;}}` et la méthode:  
`static void permutebis(B a,B b){int tmp=a.i;a.i=b.i;b.i=tmp; }`  
le morceau de code:  
`B a=new B(3); B b=new B(0); permutebis(a,b);System.out.println(a.i);`
- (a) affichera 3
  - ☐ (b) affichera 0
  - (c) provoquera une erreur à la compilation ou à l'exécution
17. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=a;((A)o).i=10;:`
- ☐ (a) est correct (il peut être compilé et exécuté)
  - (b) n'est pas correct (il ne peut pas être compilé et exécuté)
18. En java:
- ☐ (a) le type d'une expression est toujours déterminé à la compilation
  - (b) le type d'une expression peut changer au cours de l'exécution
19. Avec la classe: `class B{int i;B(int i){this.i=i;}}` le morceau de code:  
`B a=new B(3); B b=a;`  
`if(a==b) System.out.println("égal"); else System.out.println("non égal");`
- ☐ (a) affichera égal
  - (b) affichera non égal
  - (c) provoquera une erreur à la compilation ou à l'exécution
20. La commande `javac` appliquée à un programme source syntaxiquement correct:
- (a) génère un fichier exécutable sur la machine cible
  - ☐ (b) génère un fichier qui peut être interprété par la machine virtuelle java
21. En général, une *applet* java:
- ☐ (a) est un code java destiné à être exécuté sur un navigateur web
  - (b) est un code java destiné à être exécuté sur un serveur web
22. On considère la classe définie par : `class A {static int i; }`. L'exécution de:  
`A a=new A(); A b=new A(); a.i=10; b.i=5; System.out.println(a.i);`
- ☐ (a) affiche 5
  - (b) affiche 10
  - (c) provoque une erreur