

POO L2-L3 MI LI
QCM 1, Version:

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.

1. 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

2. Avec la méthode: `“static void perm(Integer i,Integer j){Integer tmp; tmp=j; j=i; i=tmp;}”` le morceau de code:
`Integer n2=10, n3=5; perm(n2,n3); System.out.println(n2);`
- (a) affichera 10
 - (b) affichera 5
 - (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;}}”` et la méthode:
`“ static void permute(B a,B b){B tmp=a;a=b;b=tmp; }”`
 le morceau de code:
`B a=new B(3); B b=new B(0); permute(a,b);System.out.println(a.i);`
- (a) affichera 3
 - (b) affichera 0
 - (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. On considère la classe définie par : `class A {int i=0; B b; }`, la classe: `B {int j=0;}` et le code `A a1=new A(); A a2=new A(); B b=new B();`
 Quelle(s) expression(s) a (ont) la valeur true:
- (a) `(a1.i != b.j)`
 - (b) `(a1.b == a2.b)`
 - (c) `(a1 == a2)`
7. En supposant que f est une méthode ayant un int en paramètre “f(1);” est:
- (a) une expression
 - (b) une instruction
8. On considère la classe définie par : `class A {static 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
9. `public static void permuter (String s1, String s2, int x1, int x2){`
`String tmp1=s1; s1=s2; s2=tmp1; int tmp2=x1; x1=x2; x2=tmp2;`
`}`
- Considérons: `String a="bon"; String b="jour"; int c=3; int d =4; permuter(a,b,c,d);`
 Quelles seront les valeurs de a,b,c,d après l'exécution de ce code?
- (a) "bon", "jour", 3, 4
 - (b) "jour", "bon", 3, 4
 - (c) "bon", "jour", 4, 3
 - (d) "jour", "bon", 4, 3

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(new Integer(2))`
- afficheront void f1(int)
 - afficheront void f1(Integer)
 - provoqueront une erreur à la compilation ou à l'exécution
11. 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(3);`
`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
12. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=a;o.i=10;;`
- est correct (il peut être compilé et exécuté)
 - n'est pas correct (il ne peut pas être compilé et exécuté)
13. Le code:
`Integer m=new Integer(2);`
`if (m==2) System.out.println("égal"); else System.out.println("non égal");`
- affichera égal
 - affichera non égal
14. On considère la classe définie par : `class A {int i=0; }`. L'exécution de:
`A a,b; a.i=10; b=a; b.i=5; System.out.println(a.i);`
- affiche 10
 - affiche 5
 - provoque une erreur
15. En java:
- toute instruction a un type
 - toute expression a une valeur
16. Avec la méthode: `static Integer f(int i){Integer n=i; return n;}` le morceau de code:
`Integer n1=2; if (2==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
17. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=a;((A)o).i=10;;`
- est correct (il peut être compilé et exécuté)
 - n'est pas correct (il ne peut pas être compilé et exécuté)
18. Après les déclarations `int i,j=0,k,l;` le morceau de code `l=1+(k=i=j+2)` :
- est une instruction qui modifie les variables i, k et l
 - est une expression de type int, qui vaut 3
 - n'est syntaxiquement pas correct

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; b.i=8; System.out.println(a.i);`
- (a) affichera 3
 - (b) affichera 8
 - (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. Le résultat de la compilation d'une applet java:
- (a) est du “byte code” java
 - (b) est un texte html
22. On considère la classe définie par : `class A {int i; static void print(){System.out.println(i); }}`. L'exécution de:
`A a=new A(); A b=a; a.i=5; b.i=10; a.print();`
- (a) affiche 5
 - (b) affiche 10
 - (c) la classe A ne peut pas être compilée

Answer Key for Exam A

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

1. 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
2. Avec la méthode: “static void perm(Integer i,Integer j){Integer tmp; tmp=j; j=i; i=tmp;}” le morceau de code:
Integer n2=10, n3=5; perm(n2,n3); System.out.println(n2);
 (a) affichera 10
(b) affichera 5
(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;}}” et la méthode:
“ static void permute(B a,B b){B tmp=a;a=b;b=tmp; }”
le morceau de code:
B a=new B(3); B b=new B(0); permute(a,b);System.out.println(a.i);
 (a) affichera 3
(b) affichera 0
(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. On considère la classe définie par : class A {int i=0; B b; }, la classe: B {int j=0;} et le code A a1=new A(); A a2=new A(); B b=new B();
Quelle(s) expression(s) a (ont) la valeur true:
(a) (a1.i != b.j)
 (b) (a1.b == a2.b)
(c) (a1 == a2)

7. En supposant que f est une méthode ayant un int en paramètre “f(1);” est:
- (a) une expression
 - (b) une instruction
8. On considère la classe définie par : `class A {static 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
9. `public static void permuter (String s1, String s2, int x1, int x2){
String tmp1=s1; s1=s2; s2=tmp1; int tmp2=x1; x1=x2; x2=tmp2;
}`
- Considérons: `String a="bon"; String b="jour"; int c=3; int d =4; permuter(a,b,c,d);`
Quelles seront les valeurs de a,b,c,d après l'exécution de ce code?
- (a) "bon", "jour", 3, 4
 - (b) "jour", "bon", 3, 4
 - (c) "bon", "jour", 4, 3
 - (d) "jour", "bon", 4, 3
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(new Integer(2))`”
- (a) afficheront void f1(int)
 - (b) afficheront void f1(Integer)
 - (c) provoqueront une erreur à la compilation ou à l'exécution
11. 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(3);
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
12. Soit la classe : `class A {public int i; }`. Le code suivant `A a=new A();Object o=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)
13. Le code:
`Integer m=new Integer(2);
if (m==2) System.out.println("égal"); else System.out.println("non égal");`
- (a) affichera égal
 - (b) affichera non égal

14. On considère la classe définie par : `class A {int i=0; }`. L'exécution de:
`A a,b; a.i=10; b=a; b.i=5; System.out.println(a.i);`
- (a) affiche 10
 - (b) affiche 5
 - (c) provoque une erreur
15. En java:
- (a) toute instruction a un type
 - (b) toute expression a une valeur
16. Avec la méthode: “ `static Integer f(int i){Integer n=i; return n;}`” le morceau de code:
“`Integer n1=2; if (2==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
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. Après les déclarations `int i,j=0,k,l;` le morceau de code “`l=1+(k=i+j+2)`” :
- (a) est une instruction qui modifie les variables i, k et l
 - (b) est une expression de type int, qui vaut 3
 - (c) n'est syntaxiquement pas correct
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; b.i=8; System.out.println(a.i);`
- (a) affichera 3
 - (b) affichera 8
 - (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. Le résultat de la compilation d'une applet java:
- (a) est du “byte code” java
 - (b) est un texte html
22. On considère la classe définie par : `class A {int i; static void print(){System.out.println(i); }}`. L'exécution de:
`A a=new A(); A b=a; a.i=5; b.i=10; a.print();`
- (a) affiche 5
 - (b) affiche 10
 - (c) la classe A ne peut pas être compilée