Faculté des Sciences
Aix*Marseille Université

Année universitaire 2020/2021

Site : ☒ Luminy ☐ St-Charles ☐ St-Jérôme ☐ Cht-Gombert ☒ Aix-Montperrin ☐ Aubagne-SATIS
Sujet de : ☒ 1er semestre ☐ 2ème semestre ☐ Session 2 Durée de l'épreuve : 1h
Examen de : L2 Nom du diplôme : Licence d'informatique
Code du module : SIN3U02 Libellé du module : Programmation 2
Calculatrices autorisées : NON Documents autorisés : NON

Inscrivez votre nom et prénom ci-dessous :
Nom :

Prénom :

Répondez directement sur le sujet en cochant ou en écrivant vos réponses aux emplacements prévus à cet effet.

1 Passage d'arguments

Soit le programme suivant

```
public class Exercice1{
    public static void main(String[] args){
    int variable1 = 2;
    Classe variable2 = new Classe(2);
    int variable3 = 4;
    variable3 = uneFonction(variable1, variable2);
    }
    public static int uneFonction(int p1, Classe p2){
    int i = p2.val;
    p1 = 4;
    p2.val = 4;
    p2 = null;
    return i;
    }
}
class Classe{
    public int val;
    public Classe(int val){
    this.val = val;
    }
}
```

Quelles sont les valeurs des expressions suivantes après l'appel à la fonction `uneFonction` ?

expression	2	4	null
variable1	☒	☐	☐
variable2.val	☐	☒	☐
variable3	☒	☐	☐

2 Interfaces

Soient les deux interfaces suivantes :

```
interface Interface1{ public void methode1();}
interface Interface2{ public void methode2();}
class Classe1 implements Interface1{public void methode1() {} }
class Classe2 implements Interface1{public void methode2() {} }
class Classe3 implements Interface1{public void methode2(){} public void methode1() {} }
class Classe4 implements Interface1{public int methode1() {} }
class Classe5 implements Interface1, Interface2{public void methode2() {  } public void methode1()
class Classe6 implements Interface1, Interface2{public void methode2() {} }
```

Dites, pour chacune des classes définies si leur définition est correcte ou incorrecte

Classe	correct	incorrect
Classe1	☒	☐
Classe2	☐	☒
Classe3	☒	☐
Classe4	☐	☒
Classe5	☒	☐
Classe6	☐	☒

3 Accessibilité

On considère le code suivant :

```
package School;
public class Subject{
    int number;
    private String name;
    public void printSubject(){ (CODE BLOC A) }
}
package School;
public class Room{
    private int roomNumber;
    protected int capacity;
    private Boolean AssignSubject(Subject X) { (CODE BLOC B) }
}
package Main-Package;
import School.*;
public class ExamRoom extends Room {
    private int examID;
    public void print() { (CODE BLOC C) }
}
```

Pour chaque bloc de code (A, B, C) dans le tableau ci-dessous cochez le ou les attributs qui sont accessibles dans ce bloc :

Code Bloc	number	name	roomNumber	capacity	examID
Bloc A	☒	☒	☐	☐	☐
Bloc B	☒	☐	☒	☒	☐
Bloc C	☐	☐	☐	☒	☒

4 Extension

Soient les deux classes suivantes :

```
class Classe1{public float a1; public void methode1() {} }
class Classe2 extends Classe1{public int a2; public void methode2() {} }
```

Dites si les déclarations et affectations suivantes sont correctes ou incorrectes

déclaration	correct	incorrect
`Classe1 a = new Classe2();`	☒	☐
`Classe2 b = new Classe1();`	☐	☒
`Classe1 f = new Classe2(); f.methode1();`	☒	☐
`Classe1 g = new Classe2(); g.methode2();`	☐	☒
`Classe2 c = new Classe2(); c.a1 = 0;`	☒	☐
`Classe1 d = new Classe1(); d.a2 = 0;`	☐	☒
`Object e = new Classe2();`	☒	☐

5 Classes Abstraites

```
public interface Big {
    public double computeSum(List<double> numbers);
}
public abstract class Bigger implements Big {/* Code à ajouter */}
public class Biggest extends Bigger {/* Code à ajouter */}
public class Main
{
    public static void main(String[] args){
        int init_value = 100;
        Bigger bigger_object = new Biggest(init_value);
        System.out.println("Mon valeur est : " + bigger_object.myValue );
        }
}
```

La méthode `computeSum()` doit être implementée dans : (Cochez la ou les bonnes réponses)

Big	Bigger	Biggest	Bigger ou Biggest	Big ou Bigger ou Biggest
☐	☐	☐	☒	☐

L'attribut `myValue` doit être declaré dans : (Cochez la ou les bonnes réponses)

Big	Bigger	Biggest	Bigger ou Biggest	main
☐	☒	☐	☐	☐

6 Mot clé `final`

```
final public class Book {
    final public String author, title;
    protected List<int> editions;
    public Book(String author, String title) {
        this.author = author; this.title = title;
        this.editions = new List<int>();
    }
    final public void addEdition(int number) { editions.add(number); }
```

```
}
```

Parmi les instruction, déclarations et expressions suivantes, lequelles sont correctes (elles ne produisent pas d'erreurs à la compilation) ? Cochez la ou les bonnes réponses.

☒ `Book myBook = new Book(" "," ");`
☐ `myBook.title = "Programmation en Java";`
☒ `myBook.addEdition(1);`
☐ `class TextBook extends Book {  ...`
☐ `@Override addEdition(int x, double y) { ...`

7 Exceptions

```
public static void editFiles(String filename) throws IOException {
    try {
                File file1 = new File(filename);
                if (file1.exists())  System.out.print("YES : ");
                else   System.out.print(" NO : ");
                Scanner input = new Scanner(file1);
                String word = input.next();
    }
    catch (NullPointerException e){ System.out.print(" (1) "); }
    finally { System.out.print(" (2) "); }
}

public static void main(String[] args){
    String str = null;
    try {
        editFiles(str);
        editFiles(" ");
    } catch(IOException  e) { System.out.print(" (3) "); }
}
```

Quel est l'affichage qui est produit par l'exécution de ce code ? Répondez directement ci-dessous

```
(1) (2) No: (2) (3)
```

8 Surcharge de méthodes

Soit l'interface suivante :

```
public interface Prog2 {
    public int computeSize(String a, String b, String c);
    public long computeSize(long a, char b);
    public double computeSize(String a, char b);
    public double computeSize(double x, Boolean b);
    public int computeSize(double a, double b, double c);
}
```

Parmi les appels suivants, dans une classe qui implémente `Prog2`, cochez ceux qui **vont provoquer** une erreur de compilation.

☒ `computeSize(new String[] {"a", "b", "c"});`
☐ `computeSize("abc",'d');`
☐ `computeSize(1, 2, 3.3);`
☐ `computeSize('p','q');`
☐ `computeSize(20*20, true);`
☐ `computeSize('a', 3.14, 5/5);`