Practice Final

```
 1  class Race{
 2          public static final int MAX = 5;
 3          public static final int MIN = 1;
 4
 5          private static int data=Race.MIN;
 6          public static int getData(){
 7                  return data;
 8          }
 9          public static void incData() {
10                  if (data < Race.MAX)
11                          data++;
12          }
13          public static void foo() throws Exception{
14                  while (data < Race.MAX)
15                          incData();
16          }
17  }
18
19  public class Car implements Runnable {
20          public void run(){
21                  Race.foo();
22          }
23          public static void main(String[] args) throws Exception{
24                  Thread t1 = new Thread(new Car());
25                  Thread t2 = new Thread(new Car());
26                  t1.start();
27                  t2.start();
28                  t1.join();
29                  t2.join();
30                  System.out.println(Race.getData());
31          }
32  }
```

1. In the above code, where might be race condition(s) occur? Specifically which line(s) of the code ?
   Yes, there are race conditions : on lines 26 and 27 there are two consecutives threads ran without join them before so t1 and t2 run at the same time.

2. Where and how should the race condition(s) be prevented?
   We should put try catch block with t1.start and t1.join then a try catch block with t2.start and t2.join.

3. What is a deadlock? Can one occur in the above code ever? Why or why not?
   ???

```
 1  class Examine{
 2          static int[] data = new int[10];
 3          public static void dance(int value){
 4                  int i=0;
 5                  while(i<data.length){
 6                          for(int j=0;j<100;j++);
 7                                  data[i] = value;
 8                  }
 9          }
10          static Runnable launch(int id){
11                  return new Runnable(){
12                          public void run(){
13                                  dance(id);
14                  }
15          }
16          public static void main(String[] args){
17                  Thread t1 = new Thread(launch(1));
18                  Thread t2 = new Thread(launch(2));
19                  t1.start();
20                  t2.start();
21                  t1.join();
22                  t2.join();
23          }
24  }
```

4. In the above code is there a race condition? If yes, what is the race condition and where exactly is it (demonstrate the race via code walk thru)?
   There is a race condition because there are two threads (t1 and t2) run at the same time (lines 19 and 20).

5. Would the code when run cause an Exception?
   Yes, because, by calling the run() method in the launch method, we try to run a thread without starting it.

```java
 1  class Grain {
 2          public String toString() { return "Grain"; }
 3  }
 4
 5  class Wheat extends Grain {
 6          public String toString() { return "Wheat"; }
 7  }
 8  class Mill {
 9          Grain process() { return new Grain(); }
10  }
11  class WheatMill extends Mill {
12          Wheat process() { return new Wheat(); }
13  }
14  public class CovariantReturn {
15          public static void main(String[] args) {
16                  Mill m = new Mill();
17                  Grain g = m.process();
18                  System.out.println(g);
19                  m = new WheatMill();
20                  g = m.process();
21                  System.out.println(g);
22          }
23  }
```

6. Is the method "process()" in WheatMill in error?
   ??? Pourquoi il n'y a pas une erreur à la ligne 19 parce que tous les Mill ne sont pas des WheatMill

```
class Egg2 {
        protected class Yolk {
                public Yolk() { print("Egg2.Yolk()"); }
                public void f() { print("Egg2.Yolk.f()");}
        }
        private Yolk y = new Yolk();
        public Egg2() { print("New Egg2()"); }
        public void insertYolk(Yolk yy) { y = yy; }
        public void g() { y.f(); }
}

public class BigEgg2 extends Egg2 {
        public class Yolk extends Egg2.Yolk {
                public Yolk() { print("BigEgg2.Yolk()"); }
                public void f() { print("BigEgg2.Yolk.f()"); }
        }
        public BigEgg2() { insertYolk(new Yolk()); }
        public static void main(String[] args) {
                Egg2 e2 = new BigEgg2();
                e2.g();
        }
}
```

7. Give the output for the above:

```
$ New Egg2()
$ BigEgg2.Yolk()
$ Egg2.Yolk.f()
```

Ne pas oublier que on appelle d'abord le constructeur des parents de la classe courante avant d'appeler le constructeur de la classe courante ; On a une variable qui appelle un constructeur dans la classe (à revoir parce que changer la ligne de la définition de la variable ne change rien : pourquoi ?)

```java
class Parcel4 {
        private class PContents implements Contents {
                private int i = 11;
                public int value() { return i; }
        }
        protected class PDestination implements Destination {
                private String label;
                private PDestination(String whereTo) {
                        label = whereTo;
                }
                public String readLabel() { return label; }
        }
        public Destination destination(String s) {
                return new PDestination(s);
        }
        public Contents contents() {
                return new PContents();
        }
}
public class TestParcel {
        public static void main(String[] args) {
                Parcel4 p = new Parcel4();
                Contents c = p.contents();
                Destination d = p.destination("Tasmania");
                Parcel4.PContents pc = p.new PContents();//ERROR
        }
}
```

8. The above class has an error on the line indicated. Explain why there is an error.
   (revoir les inner et outer classes statiques ou non :) On essaye d'accéder au constructeur d'un objet qui est une classe privée sans utiliser un objet de cette classe.
   Deux solutions :

   - mettre PContents en statique puis Parcel4.PContents pc = new Parcel4.PContents();
   - créer un objet Contents à partir de la méthode contents() puis cast l'objet en Parcel4.PContents

```java
public class Wrapping {
        private int i;
        public Wrapping(int x) { i = x; }
        public int value() { return i; }
}

public class Parcel {
        public Wrapping wrapping(int x) {
        //return an anonymous inner class object of Wrapping type
        //with overloaded method "public int value (){ return 47*i;}"
        }
        public static void main(String[] args) {
                Parcel p = new Parcel();
                Wrapping w = p.wrapping(10);
        }
}
```

9. In the above classes provide the missing code.

```java
// Passe
```

10. Given an ArryList is-a List and a List is-a Collection is the following true? ArrayList<String> is-a Collection<String>

11. Given interface FooBar<X,Y> extends Silly<X> which of the following ARE subtypes of Silly<String>? b and d (X has to be String and Y has to be not String)

    (a) FooBar<String, String>
    (b) FooBar<String, Integer>
    (c) FooBar<Integer, String>
    (d) FooBar<String, Exception>
    (e) FooBar<Integer, Integer>
    (f) FooBar<Exception, Integer>

12. Given

    static <T>T pick(T a, T b){return b}

    Is the following an error or not? If an error, explain why, if not give the return type.

    Collection c = pick(new Set<String>(), new Stack<String>());

    There is an error because the two arguments of pick() method have to be the same type but Set<> and Stack<> aren't the same type.

13. If the code below gives an error explain why, if not explain why.

```
public static void addNumbers(List<? super Integer> list){
        for (int i=1; i<10; i++){
                list.add(i);
        }
}
// in other code:
addNumbers(new ArrayList<Number>());
```

The code above doesn't give an error because an ArrayList is a List and ? super Integer includes Number.

14. If the code below gives an error explain why, if not explain why.

```
void swapFirst(List<? extends Number>listA, List<? extends Number> listB){
        Number temp = listA.get(0);
        listA.set(0,listB.get(0));
        listB.set(0,temp);
}
```

There is no compile-time error because we only deal with Number objects.

```
class Example{
        public void open() throws FileNotFoundException{
                System.out.println("attempting to open file");
                throw new FileNotFoundException();
        }
        public void close() throws CloseException {
                System.out.println("attempting to close file");
                throw new CloseException();
        }
        public static void main(String[] args) throws Exception{
                Example e = new Example();
                try{
                        e.open();
                        System.out.println("after opening file");
                }finally{
                        System.out.println("finally");
                        e.close();
                        System.out.println("after closing file");
                }
                System.out.println("end of program");
        }
}
```

15. Give the output. State any exception(s) that are displayed on exit.

```
$ attempting to open file
$ finally
$ attempting to close file
$ error CloseException
```

```
class LanguageException extends Exception{}
class JavaException extends LanguageException{}

public class Test {
        public void a() throws LanguageException{
                throw new LanguageException();
        }
        public void b() throws JavaException{
                throw new JavaException();
        }
        public static void main(String[] args){
                Test t = new Test();
                try{
                        t.a();
                        t.b();
                }
                catch(LanguageException l){}
                catch(JavaException j){}
                System.out.println("finished main");
        }
}
```

16. Give the output. State any exception(s) that are displayed on exit.

```
// Reponse a la question 16
$ finished main
```

```
public class Out {
        int x;
        static class In {
                public void setX(int value){
                        x = value;
                }
        }
        public static void main(String[] args){
                //your code goes here
        }
}
```

17. What is the error in the above class? Why?
    The error is because we try to modify the value of Out.x with the Out.In.SetX() method. Why?

18. Give the code to create an "In" object in main()

```
public static void main(String[] args){
        Out.In in = new Out.In();
}
```

```java
class Cat {
	Kitten k = new Kitten();
	public Cat(){
		System.out.println("cat");
	}
	class Kitten{
		public Kitten(){
			System.out.println("kitten");
		}
	}
}
public class Lion extends Cat {
	public Lion(){
		System.out.println("Lion");
	}
	class Kitten {
		public Kitten(){
			System.out.println("young Lion");
		}
	}
	public static void main(String[] args){
		new Lion();
	}
}
```

19. Give the output

```
// Reponse question 19 (code au dessus)
$ kitten
$ cat
$ Lion
```

```java
class Cat {
	Kitten k;
	public Cat(){
		System.out.println("cat");
	}
	class Kitten{
		public Kitten(){
			System.out.println("kitten");
		}
	}
	public void produce(Kitten kk){
		k = kk;
	}
}
public class Lion extends Cat {
	public Lion(){
		System.out.println("Lion");
		produce(new Kitten());
	}
	class Kitten {
		public Kitten(){
			System.out.println("young Lion");
		}
	}
	public static void main(String[] args){
		new Lion();
```

```
        }
}
```

20. Give the output or if there is an error, fix it and give the output.

```
// Reponse question 20 (code au dessus)
$ cat
$ Lion
$ young Lion
// Faux : on ne peut pas mettre un Lion.Kitten dans un Cat.Kitten
```

To fix it, we have to make the Cat.Kitten class static, then we create a new Cat.Kitten() in Lion constructor, then the output is cat, Lion, kitten

```
class A {
        private int x;
        public void setZ(int zz){
                z = zz;
        }
        class B{
                private int y;
                class C{
                        private int z;
                        public void setX(int xx){
                                x = xx;
                        }
                }
        }
}
```

21. Examine the code above. Is there an error? If so what is the error and why? If not, explain. First, in class A, the method setZ() set z attribute to zz value but there is no z atribute and in setX() of class C, we set x attribute to xx value but x is a private attribute of class A so it's not accessible from class C.

```
class A{
        class B{
                class C{}
        }
        public static void main(String[] args){
                //code
        }
}
```

22. Give the code necessary to create a C object in main().

```
public static void main(String[] args){
        // Passe
}
```

```
class X {
        int z = 5;
        static class Y{
                public int getZ(){return z;}
        }
        public static void main(String[] args){
```

```
                        //code
            }
}
```

23. What is the an error in the code above?
    We can't access to z attribute in class Y.

24. If the error was removed in the above code, give the code to create a Y object.

```
public static void main(String[] args){
                        X x = new X();
                        int value = X.Z.getZ(x);
}
```

```java
class X{
        int y;
        public void foo(){
                for(int i=0; i<10; i++){
                        if (y < 5)
                                y++;
                }
        }
}
```

25. For the above code give the places where a race condition exists.
The race condition exist on the if(y<5) because we modify the value of y. If two threads access at the same time to the y value and y==4, y will be incremented by the number of current threads on it. So, at the end of the calls of foo, y might be greater or equal to 5.

```java
class A implements Runnable {
        ReentrantLock lock = new ReentrantLock();
        List s ;
        public A(List store){
                s = store;
        }
        public void run(){
                for(int i=0;i<10; i++){
                        if(lock.trylock())
                                s.add(i);
                }
        }
}

class B implements Runnable{
        ReentrantLock lock = new ReentrantLock();
        List s;
        public B(List store){
                s = store;
        }
        public void run(){
                for(int x = 20; x<30; x++){
                        if(lock.tryLock())
                                s.add(x);
                }
        }
}
public class Test {
        public static void main(String[] args){
                List store = new LinkedList();
                A a = new A(store);
                B b = new B(store);
                new Thread(a).start();
                new Thread(b).start();
                Thread.sleep(2000);
        }
}
```

26. Does the above code protect the shared List? Why or why not?
Passe

```
public class LTest {
        public static void main(String[] args){
                new Thread(new Runnable(){
                        public void run(){
                                System.out.println("hello world!");
                        }
                }).start();

        }
}
```

27. Write the above using a Lambda expression.

```java
public class Table{
        enum TYPE {MULT, ADD}
        public void display(int start, int end, TYPE t){
                for (int i = start; i<end; i++){
                        for (int j = start; j <end; j++){
                                if (t == TYPE.MULT)
                                        System.out.printf("%3d ",(i*j));
                                else
                                        System.out.printf("%3d ",(i+j));
                        }
                        System.out.println("");
                }
        }
        public static void main(String[] args){
                Table t = new Table();
                t.display(1,10,TYPE.MULT);
        }
}
```

28. Modify to use a Lambda expression and give the expression to get the same output as the code above with your new display() method.