

Q1. Lab09: Day provides the clone() method. SystemDate inherits Day. Therefore clone() is available for SystemDate:

```
@Override
public Day clone() {
    Day copy=null;
    try {
        copy = (Day) super.clone();
    } catch (CloneNotSupportedException e) {
        e.printStackTrace();
    }
    return copy;
}
```

```
public Member(String id, String name) {
    this.id = id;
    this.name = name;
    this.joinDate = SystemDate.getInstance().clone();
    Club.getInstance().addMember(this);
    //System.out.println(joinDate.getClass());
    //Output: class SystemDate
}
```

Problem: As a singleton we should avoid to create one more object of the SystemDate class. Indeed, for the dateSetup field inside Team, we only need a Day object, not a SystemDate object

Solution 1: We provide a method, copyDay() of SystemDate, that copies and returns a Day object. (However, the drawback is, the clone() method from the Day class is still available.)

Solution 2: Change the design of SystemDate such that it contains the Day field instead of inheriting Day. Therefore, it will not have the clone() method. Try!

Solution 3: We implement the .clone() method of SystemDate that creates and return a Day object. Try!

Q2. Complete the following summary about Exception Handling

[Contents extracted from <http://docs.oracle.com/javase/tutorial/essential/exceptions/summary.html>]

A program can use exceptions to indicate that an error has occurred.

To _____ an _____, use the throw statement and provide it with a error message / an exception object.

An exception object is a descendant of _____ which aims at _____ about the specific error that occurred.

[Given words for blanks]
 throw, exception, Throwable
 providing information
 catch the exception
 a throws clause
 the try, catch, and finally blocks
 exception handler

Checked exceptions are exceptions which the compiler will reinforce proper handling in your code.

A method that throws a checked / unchecked exception must either

_____ or include _____ in its

declaration.

A program can catch exceptions by using a combination of _____.

- The **try** block identifies a block of code in which an exception can occur / be thrown.
- The **catch** block identifies a code block, known as an _____, that can handle a particular type of exception.
- The **finally / try** block identifies a block of code that is guaranteed to execute, and is the right place to close files, recover resources, and otherwise clean up after the try-catch / finally block.

The class / instance of the exception object indicates the type of exception thrown. By convention, the exception object contains at least an error message.

■ The Catch-or-Declare Rule

- If an exception may be thrown inside a method, the method may deal with it by
 1. placing the concerned code within a try-block, and handle in a catch-block,
- or
- 2. declaring it

```
public void methodX()
{
    ..
    try { ... //code which may throw exception
    }
    catch (.. e) {catch and handle the exception
    }
}
```

```
public void methodX() throws ..
{
    ..
}
```

- The compiler reinforces the Catch-or-Declare Rule on the following exceptions [Known as "checked exceptions"]:
 For other exceptions/errors, it is okay whether we deal with them or not. [Known as "unchecked exceptions"]

Must Catch-or-Declare ["Checked exceptions"]

```

graph TD
    Throwable --> Error
    Throwable --> Exception
    Exception --> IOException
    Exception --> RuntimeException
            
```

www.cs.cityu.edu.hk/~helena Topic06 25