CP 213 Lecture 11 start

Deep vs Shallow copy

Deep copy is a copy that w/ one exception (Immutable objects) has no references in common w/ original

Any copy that isn't a deep copy is a shallow copy

Pitfalls - ~~Another~~ previous type of privacy leak

Is also caused by the following

Code
```
public Date getBirthDate() {
    return born; } // dangerous
```

Instead

Code
```
public Date getBirthDate() {
    return new Date(Born); } // correct
```

```
Stack (Stack orig) {
    this.top = orig.top; } // shallow copy
```

Instead

```
Stack (Stack orig) {
    this.top = new Node (orig.top); } // must make copy construction
```

~~This element~~

```
Node (Node other) {
    this.element = orig.element;
    if (other.next != null) {
        this.next = null; }
    else {
        this.next = new Node (other.next); } }
Stack (Stack orig) {
    if (orig.top != null) {
        this.top = new Node (orig.top); }
    else {
        this.top = null; } }
```

Lecture 11 (continued)

Introduction To Inheritance

↳ Allows a new class to be made based on existing class

↳ new class is derived/child/sub class

↳ original class is base/parent/super class

A derived class automatically has all the instance variables & methods that the base class has & can have additional methods &/or instance variables

Lecture 12 start Oct 5th 2016 CE

Override → completely replaces definition

overload → adds to definition

↳ Several methods same name but different arguments

Cannot make public method private in subclass

Can make private method public in subclass

Cannot override a "final" method

↳ but can overload it

Subclass constructor

```
public HourlyEmployee() {
    super()
    wageRate = 0
    hours = 0
}
```

→ can't have both "super" & "this" in same definition

{ this("no name", new Date("Jan", 1, 1000)); }

is same as calling 2 parameter constructor in a method

{ super("no name", new Date("Jan", 1, 1000)); } → calls base constructor

```
public HourlyEmployee (String theName, Date theDate,
                       double theWageRate, double theHours)
{ super(theName, theDate);
  if ((theWageRate >= 0) && (hours >= 0)) {
    wageRate = theWageRate
    hours    = theHours
```

Lecture start Monday oct 17th 2016 EE