CSE 8B Today

PSA4: Start now!

Why objects?
Why memory models?

Make sure you review them for next exam (Tuesday next week)

Nim.java (posted on the website)
A VERY GOOD place to start for PSA4

main scope

Nim object scope

board null

Nim constructor scope

board

0 0 0 0 0

game1
Finish the code in the following method named `onlyLettersUp` that takes a `String s` as input and returns a new `String` object where all letters in `s` have uppercase i.e., uppercase letters stay the same, while lower case letters have become upper case letters. Non-alphabetic characters in the string are replaced with a dot character (.)

Here are a couple examples. If you are at all unsure about how this method should work, please ask.

- `onlyLettersUp ( "You Got It!!" )` returns the string "YOU.GOT.IT.."
- `onlyLettersUp ( "U_C_S_D.2014" )` returns the string "U.C.S.D....."

You need to USE a `char[] chars` to access the elements from the String at some point.

```java
public String onlyLettersUp ( String s ) { 
    char[] chars = . . . 

    // WHAT ARE THE STEPS OF MY ALGORITHM? ("plan" or "recipe" . . .)

} 
```
More on Variable Scope + PSA4 hints

ConnectFourBoard b = new ConnectFourBoard(6,7);
b.addMove(0, 'X');
b.addMove(0, 'O');
b.addMove(0, 'X');
b.addMove(3, 'O');
b.addMove(4, 'O');
// cheating by letting O go again!
b.addMove(5, 'O');
b.addMove(6, 'O');
System.out.println(b);

| | | | | | | |
| | | | | | | |
| | | | | | | |
|X| | | | | | |
|O| | | | | | |
|X| | O|O|O|O|
---------------
0 1 2 3 4 5 6
PSA4: HINT about printing the Board . . .

public String toString() // toString from Nim.java
{
    String toReturn = new String();
    for (int i = this.board.length-1; i >= 0; i-- )
    {
        toReturn += "Row " + i + " : " ;

        for ( int sticks = 0; sticks < this.board[i]; sticks++ )
        {
            toReturn += "| " ;
        }
        toReturn += ( "\n" ) ;
    }

    return toReturn;
}

In PSA4: MAKE SURE you ACCESS the DATA in this.board[i] . . .
To print X, O or ` `
public class Nim {
    private int[] board;
    public Nim() {
        int[] board = new int[4];
        board[0] = 7;
        board[1] = 5;
        board[2] = 3;
        board[3] = 1;
    }

    public static void main(String[] args) {
        Nim game1 = new Nim();
        Nim game2 = new Nim();
        game1.board[2] = 100;
        game1 = game2;
        System.out.println(game1.board[2]);
    }
}
public class Nim {
    private int[] board; //instance variable

    public Nim() {
        int[] board = new int[4]; //local variable
        board[0] = 7;
        board[1] = 5;
        board[2] = 3;
        board[3] = 1;
    }
}

The scope of a variable is the part of a program where it can be accessed:

• Variables declared inside methods are called local variables. They exist ONLY inside the method they were declared. Parameters are a special type of local variable.
• Variables declared outside methods (inside classes) are member variables or instance variables. They exist throughout the whole class definition (in all class methods).
• When local variables have the same name as member variables, they “take over” (they are said to “shadow” member variables). Here the local variable board shadows the member variable board.
public class Nim {
    private int[] board;
    public Nim()
    {
        int[] board = new int[4];
        this.board[0] = 7;
        this.board[1] = 5;
        this.board[2] = 3;
        this.board[3] = 1;
    }

    public static void main( String[] args )
    {
        Nim game1 = new Nim();
        Nim game2 = new Nim();
        game1.board[2] = 100;
        game1 = game2;
        System.out.println( game1.board[2] );
    }
}
Check out Nim.java as an example before you start PSA4

AND TRY DO THE FOLLOWING review exercises
(exam 2 in class next week!)
// Somewhere in a class
public static void main( String[] args )
{
    int myInt = 5;
    int[] myArray = new int[4];
    myArray[3] = myInt;
}

Draw the memory model for the code above
// Somewhere in a class
public static void main( String[] args )
{
    int myInt = 5;
    int[] myArray = new int[4];
    int[] anotherArray = myArray;
    myArray[3] = myInt;
    System.out.println( anotherArray[3] );
}

Draw the memory model for the code above

What does this code print?
A. 3       B. 0       C. 5       D. “myInt”
// Somewhere in a class
public static void main( String[] args )
{
    int myInt = 5;
    int[] myArray = new int[4];
    int[] anotherArray = myArray;
    myArray[3] = myInt;
    myArray = new int[5];
    System.out.println( anotherArray[3] );
}

Draw the memory model for the code above

What does this code print?
A. 3   B. 0   C. 5   D. “myInt”
public class Nim {
    int numPiles;
    public Nim( int piles ) { numPiles = piles; }
    public static void main( String[] args ) {
        int myInt = 5;
        Nim n = new Nim( myInt );
        System.out.println( n.numPiles );
    }
}

What does this code print?
A. “n”  B. 0  C. 5  D. “myInt”  E. There is an error
public class Nim {
    int numPiles;
    public Nim( int piles ) {
        numPiles = piles;
        piles = 3;  }
    public static void main( String[] args ) {
        int myInt = 5;
        Nim n = new Nim( myInt );
        System.out.println( n.numPiles );
    }
}

What does this code print?
A. 3        B. 0        C. 5        D. “myInt”        E. There is an error
What does this code print?
A. 3 B. 0 C. 5 D. “myInt” E. There is an error
What does this code print?
A. 3     B. 0     C. 5     D. 2     E. There is an error
public class Nim {
    int[] board;

    public Nim( int[] board ) {
        this.board = board;
        board = new int[3];
    }

    public static void main( String[] args ) {
        int[] board = {1, 2, 3};
        Nim n = new Nim( board );
        board[2] = 5;
        System.out.println( n.board[2] );
    }
}

What does this code print?
A. 3    B. 0    C. 5    D. 2    E. There is an error
1) Draw the memory model for the code

### Review problem: Classes, objects and scope

```java
public class Nim {
    private int numPiles;
    private int[] board;
    public Nim( int[] board ) {
        numPiles = board.length;
        this.board = new int[numPiles];
        this.board = board;
    }
    public static void main( String[] args ) {
        int[] myBoard = {1, 3, 5, 7};
        Nim n = new Nim( myBoard );
        myBoard[1] = 10;
        System.out.println( n.board[1] );
    }
}
```

2) Rewrite the constructor so that the change to myBoard in main does not change the board stored in the Nim object
public class Student {
    int numGrades;
    char[] grades;

    public Student( char[] grades ) {
        numGrades = grades.length;
        grades = grades;
    }
    public static void main( String[] args ) {
        char[] myGrades = {'B', 'B', 'C'};
        Student s = new Student( myGrades );
        myGrades[1] = 'A';
        System.out.println(s.grades[1]);
    }
}

What is printed by the this code?
A. ‘A’
B. ‘B’
C. ‘C’
D. Nothing, there is an exception in the constructor
E. Nothing, there is an exception in main (that does NOT occur in the constructor)
public class Student {
    int numGrades;
    char[] grades;

    public Student( char[] grades ) {
        numGrades = grades.length;
        this.grades = new char[this.numGrades];
        this.grades = grades;
    }

    public static void main( String[] args ) {
        char[] myGrades = {'B', 'B', 'C'};
        Student s = new Student( myGrades );
        myGrades[1] = 'A';
        System.out.println( s.grades[1] );
    }
}

What is printed by the this code?
A. ‘A’
B. ‘B’
C. ‘C’
D. Nothing, there is an exception in the constructor
E. Nothing, there is an exception in main (that does NOT occur in the constructor)
What is printed by this code?

A. 0  
B. 3  
C. 4  
D. Nothing, there is an exception in the constructor  
E. Nothing, there is an exception in main (that does NOT occur in the constructor)