COMP1409: Introduction to Software Development I

Mike Mulder (mmulder10@bcit.ca)

Week 8

Agenda

- Week 7 Review
- Quiz
  - Quiz 6
  - Review Answers
- Logistics
  - Assignment 2
  - Final Exam
- Lesson 8
  - Arrays
  - Arrays in while loops
- Lab 8

Week 7 Review Questions

What is a reference?

An address to an object in memory

What's the difference between `if(input == "bye") versus if(input.equals("bye"))?`

The first tests identity (the references are the same) and the second tests for equality. This applies to object types only.

For primitive types, the == tests for the equality of the values.

Week 7 Review Questions

What does the this keyword provide?

It is the address of the current object (i.e., it refers to the current object)

What does null mean?

An unset reference that refers to nothing

Week 7 Review Questions

```
String b1 = “something1”;
String b2 = “something2”;
String b3 = b2;
String b4 = null;
String b5 = “something different”;

System.out.println(b1);          // What does this output?
System.out.println(b1 == b2);    // What does this output?

b1 = b5;
b3 = b4;
b2 = b3;

System.out.println(b1);          // What does this output?
System.out.println(b2);          // What does this output?
System.out.println(b3);          // What does this output?
```

something1

false

something different

null

null

How many objects did we have at the beginning (i.e., after the variables were declared)?
How many object remain at the end (i.e., how many references to objects are left)?

Quiz 7

Closed book, laptop, phone, etc.

You have a maximum of 20 minutes to complete

Raise your hand when you are done, and I will retrieve your paper

We will review the answers afterwards

Logistics

Assignment 2 – Due November 16th at midnight

- Make sure all the unit tests pass

Final Exam

- Dec. 8 – In Class
- Written and coding parts. Coding part is similar to assignments – given a unit test as input and need to code the class(es)

Assignment 3

- Composition and ArrayLists
- Due the week before the Final Exam

Logistics – Remaining Classes

Week	Date	Topics	Comments
9	Nov. 3	Arrays while loops	Quiz 7
10	Nov. 10	Remembrance Day – No Class	
11	Nov. 17	More arrays for loops	Quiz 8, Assignment 2 Due
12	Nov. 24	ArrayList class Enhanced for (foreach) loop	Quiz 9
13	Dec. 1	Iterators	Quiz 10
Course Evaluation: To be conducted online during Session 11 prior to the class break.			
14	Dec. 8	Final Exam	Assignment 3 Due

Session 8 Learning Outcomes

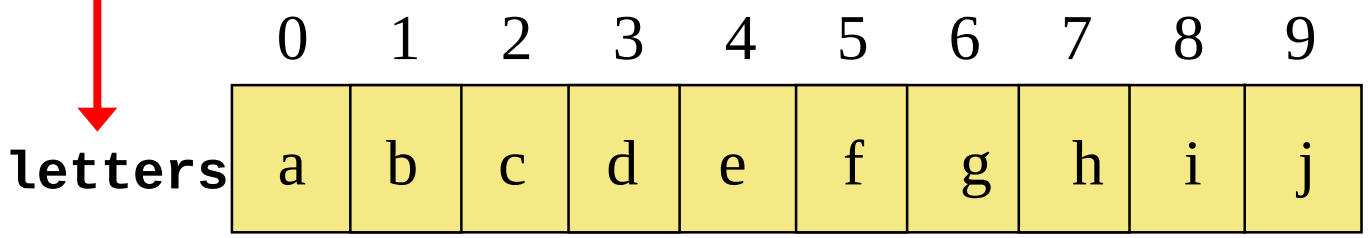
1. Arrays
2. Arrays with while loop

Arrays

An Array is an ordered list of values:

The entire Array has a single name

Each element has an integer *index*, *(always, automatically)*



An Array of size n is always indexed from zero to n-1

The Array above contains 10 chars that are indexed from 0 to 9

Array Terminology

- Element – A single value in an array (can be a primitive type or an object)
- Length – The total number of elements than can exist in an array (note: for objects they can be null)
- Index – The number that identifies an element in an array. Starts at 0 and ends at length-1
- Indices – The plural of index

Arrays

- A single value (“element”) in an Array is referenced using the Array name and its index in square brackets:

  `letters[2]`

  refers to the value `c` (the 3rd value in the Array; since the 1st value is letters**[0]**)

- letters is an Array

- letters[2] is a char

Array constraints

- An Array's length is fixed. If you set it to 12, it is always 12…never higher or lower; even if it's "empty" the size would be 12; even if you needed 13, the size remains at 12
- Any single Array can contain just one type. For example, you can have an Array of Strings, but then all the elements inside *must* be Strings. You cannot mix and match types within one Array.

Using an Array

- Arrays can contain either primitives or object references
- Square-bracket notation is used to access an Array element:

```
letters[index]
```

- Elements are used like ordinary variables:
  1. On the left of an assignment:

```
letters[3] = ...;
```

  2. In an expression:

```
String s = letters[0] + letters[1];
```

Arrays and Magic Numbers

- It is best practice to use a constant to define the size of an array if the size is known in advance

```
private static final int NUM_INSTRUCTORS = 10;

…

instructorNames = new String[NUM_INSTRUCTORS];
```

- If you are accessing a specific element, It is best practice to use a constant to define the index of that element

```
private static final int MIDTERM_EXAM = 5;

private static final int FINAL_EXAM = 10;

…

gradePercentages[MIDTERM_EXAM] = 95;

gradePercentages[FINAL_EXAM] = 88;
```

Steps to using an Array

```
private int[]          gradePercentages;
private String[]       studentNames;

...

gradePercentages = new int[24];
studentNames = new String[CLASS_SIZE];

...

gradePercentages[FINAL_EXAM] = 90;
return gradePercentages[FINAL_EXAM];
System.out.println(studentNames[index]);
```

1. DECLARE

2. INITIALIZE

3. USE

Array literals

- We can declare and initialize a String literal in one step:

```
String myFirstName = “Java lover”;
```

- We can also **declare** and **initialize** an Array literal in one step too, using {curly braces}:

```
private char[] vowels = { ‘a’, ‘e’, ‘i’, ‘o’, ‘u’ };
```

- Note: Array literals can only be used in initializations

Array literals

```
class Province{

    private String[] cities;

    public Province(){

        String homeTown = "vancouver";
        String[] cities2 = {homeTown, null, null, "calgary", "ottawa"}; // Array literal; length is 5

        cities = new String[4];      // there are four Strings in the cities Array

        cities[1] = "edmonton";
        cities[1] = "calgary"; // no more edmonton

        cities[0] = "vancouver";
        cities[3] = "saskatoon";
    }
}
```

Array length

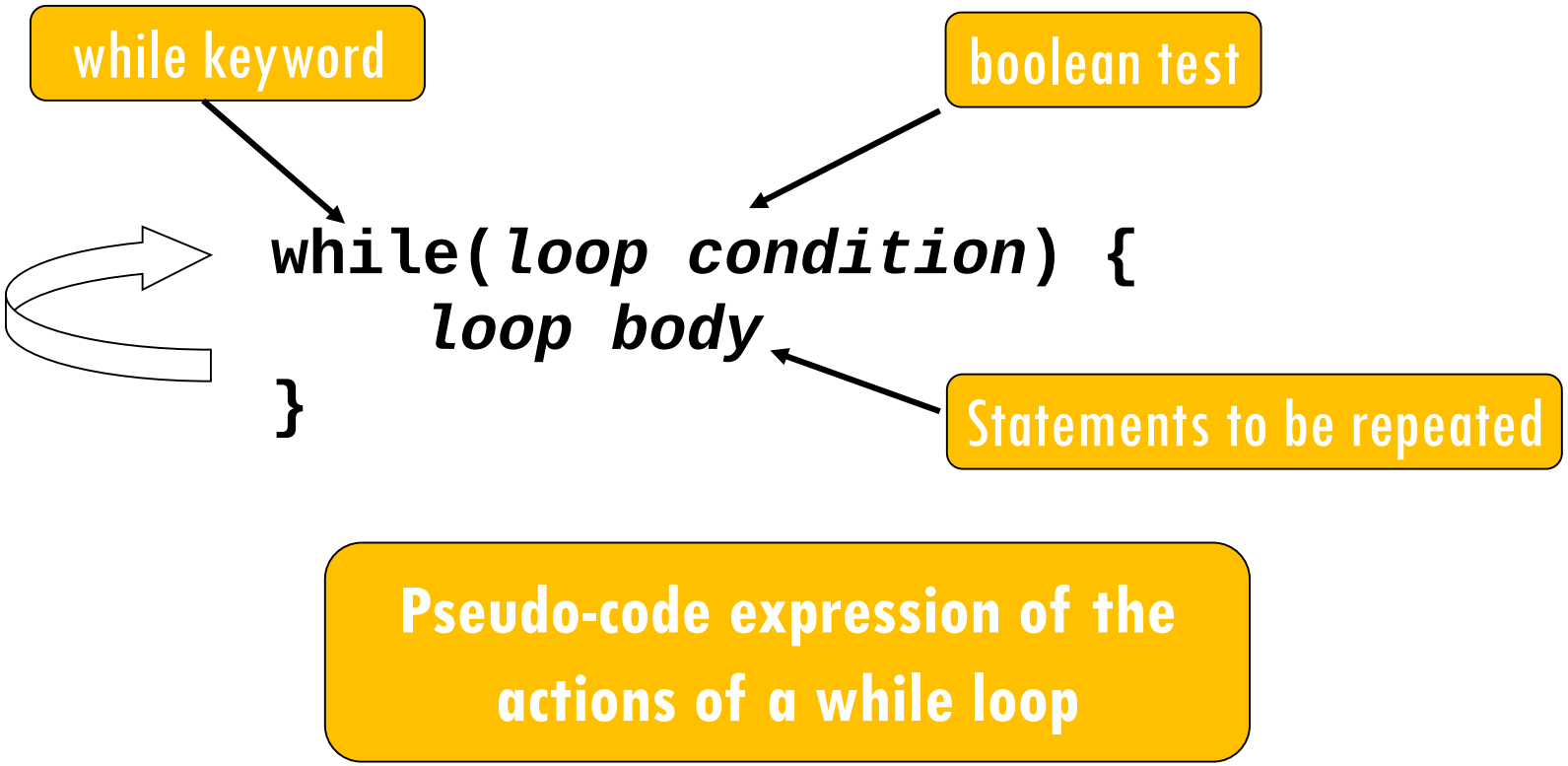
```
private char[] vowels= { ‘a’, ‘e’, ‘i’, ‘o’, ‘u’ };
int n = vowels.length; // n = 5
```

- We can determine the size of the Array using the `length` operator
- Note there are no parentheses ( )
- `length` is not a method
- `length` is a public final int property

Loops

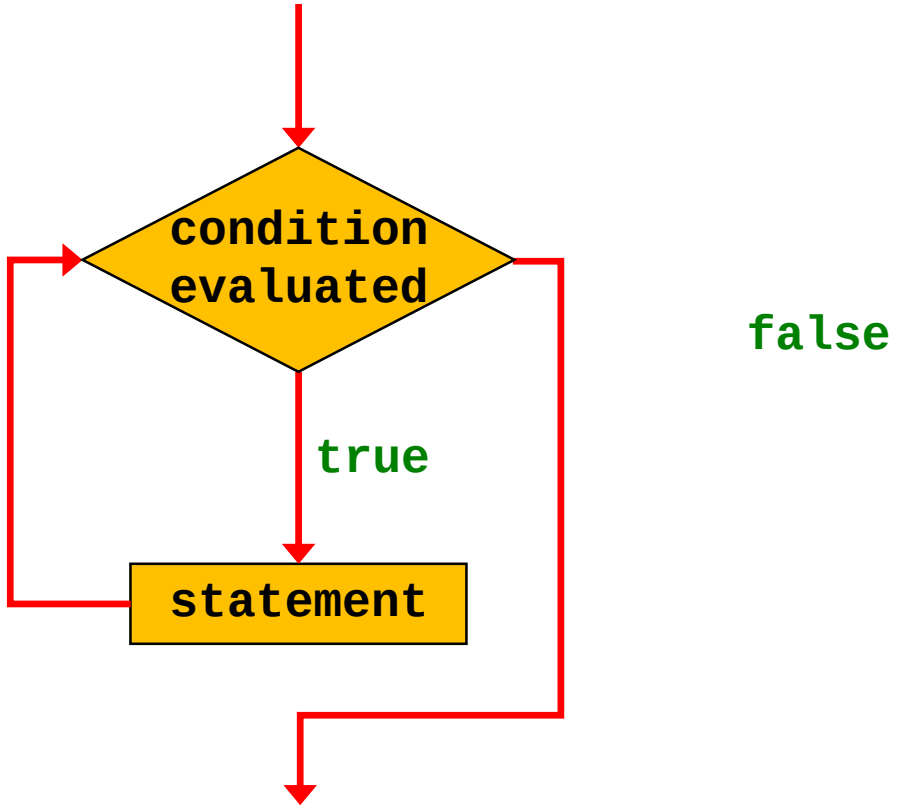
- *Repetition statements* allow us to execute a statement multiple times
- Like “if” statements, loops are controlled by boolean expressions
- Java has several kinds of repetition statements, such as `while loops, for loops, and do-while loops`
- Let’s look at the `while` loop (again).

while Statement



"while the condition is true, do the things in the loop body."

while Statement



while Statement

```
while(condition)
{
    statement 1;
    statement 2;
    etc…
}
```

1. If the condition is true, the statements are executed
2. Then the condition is evaluated again, and if it is still true, the statements are executed again
3. Repeat until the condition becomes false

while Statement

```
// Print even numbers from 2 to 30

int index = 2;
while(index <= 30) {
    System.out.println(index);
    index = index + 2;
}
```

while Statement

```
// Sum even numbers from 2 to 30
int index = 2;
int sum = 0;
while(index <= 30) {
    sum += index;
    index = index + 2;
}
System.out.println(“Sum = ” + sum);
```

while Statement With an Array

```
private int[] numbers = {2, 4, 6, 8};
int index  = 0;
int sum    = 0;
while(index < numbers.length) {
    sum += numbers[index];
    index++;
}
System.out.println(“Sum = ” + sum);
```

while Statement With an Array

```
public void printNames(){
    String[] peopleNames = {"donald", null, "hilary"};

    int i = 0;

     if(peopleNames != null){
          while(i < peopleNames.length){
                if(peopleNames[i] != null){
                     System.out.println(peopleNames[i]);
            }
           }
        i++;
    }else{
         System.out.println(“no array!!!”);
    }
}
```

Review

1. How do you declare an array variable?

```
private int[]      gradePercentages;
private String[]   studentNames;
```

2. How do you initialize an array?

```
gradePercentages = new int[24];
studentNames = new String[CLASS_SIZE];
```

3. How do you use an array?

```
gradePercentages[FINAL_EXAM] = 90;
return gradePercentages[FINAL_EXAM];
System.out.println(studentNames[index]);
```

Review

1. What is the index for the first element in an array?

```
0
```

2. What is the index of the last element in an array?

```
length-1 (i.e., one less than the maximum number of
elements in the array)
```

3. What happens if you try to access an element with a negative index or an index equal to or greater than the maximum number of elements (i.e., >= length)?

```
Array index out of bounds exception
```

Question

What is a limitation of arrays?

The size has to be known upfront (i.e., when we initialize and before we use it).

We will address that next class with an ArrayList.

Lab 8 and Next Week

Lab 8

Part A - In-Class Show me how far you got before you leave.

Part B – Submit to D2L by Thursday at midnight.

Next Week

Remembrance Day Holiday

Following Week

More arrays and loops