CSE115 / CSE503
Introduction to Computer Science I

Dr. Carl Alphonce
343 Davis Hall
alphonce@buffalo.edu

Office hours:
Thursday 12:00 PM – 2:00 PM
Friday 8:30 AM – 10:30 AM
OR request appointment via e-mail
Turn off and put away electronics:

cell phones
pagers
laptops
tables etc.
Where we’ve been
control structures
collections

Today
collections & control structures:
iterating through a collection of values

Where we’re heading
collections
search
COLLECTIONS and CONTROL STRUCTURES
PROGRAM:

```java
ArrayList<String> words;
words = new ArrayList<String>();

words.add("cat");
words.add("dog");
words.add("rabbit");

for (String w : words) {
    System.out.println("A " + w + " is a good pet");
}
```

OUTPUT:

```
A cat is a good pet
A dog is a good pet
A rabbit is a good pet
```
for ( <stmt_i> ; <expr> ; <stmt_u> ) <stmt_b>
PROGRAM:

ArrayList<String> words;
words = new ArrayList<String>();

words.add("cat");
words.add("dog");
words.add("rabbit");

for (int i=0; i<words.size(); i=i+1) {
    String w = words.get(i);
    System.out.println("A "+w+" is a good pet");
}

OUTPUT:

A cat is a good pet
A dog is a good pet
A rabbit is a good pet
Collection\(<E>\) has this method:

\[
\text{public Iterator}\(<E>\) \text{ iterator()}
\]

returns an iterator over elements currently in collection
An iterator guarantees to visit all members of a collection exactly once between the time it is created to the time its `hasNext()` method returns false.
Three methods; we focus on these two:

public boolean hasNext()
    returns true if there are items of the underlying collection that still need to be visited
    returns false if there are no unvisited items

public E next()
    returns one of the unvisited items, if any
    should not be called if hasNext() returns false
hasNext()'s return type is boolean

call to hasNext() can be used to control while loop
PROGRAM:

```java
ArrayList<String> words;
words = new ArrayList<String>();

words.add("cat");
words.add("dog");
words.add("rabbit");
Iterator<String> it = words.iterator();
while (it.hasNext()) {
    String w = it.next();
    System.out.println("A " + w + " is a good pet");
}
```

OUTPUT:

A cat is a good pet
A dog is a good pet
A rabbit is a good pet
A String object holds a sequence of char values.

The `length()` method returns the number of characters in the String.

```
"dog".length() \rightarrow 3
```

```
"rabbit".length() \rightarrow 6
```
PROGRAM:

```java
ArrayList<String> words;
words = new ArrayList<String>();

words.add("cat");
words.add("dog");
words.add("rabbit");
for (int i=0; i<words.size(); i=i+1) {
    String w = words.get(i);
    System.out.println("The length of "+w+" is "+w.length());
}
```

OUTPUT:

```
The length of cat is 3
The length of dog is 3
The length of rabbit is 6
```
A String object holds a sequence of char values.

Each char value has a position (index) within the String. An index is an integer between 0 and the length() of the string, inclusive at 0, exclusive at length().

The charAt(int) method of String returns the char at the given index:

“dog”.charAt(0) → ‘d’
“dog”.charAt(2) → ‘g’
public void characterInString(String s) {
    System.out.println("The String " + s + " consists of these characters: ");
    for (int i=0; i<s.length(); i=i+1) {
        char ch = s.charAt(i);
        System.out.println("    '"+ch+"'");
    }
}
PROGRAM:

```java
ArrayList<String> words;
words = new ArrayList<String>();

words.add("cat");
words.add("dog");
words.add("rabbit");
for (int i=0; i<words.size(); i=i+1) {
    String w = words.get(i);
    characterInString(w);
}
```
OUTPUT:
The String cat consists of these characters:
  'c'
  'a'
  't'

The String dog consists of these characters:
  'd'
  'o'
  'g'

The String rabbit consists of these characters:
  'r'
  'a'
  'b'
  'b'
  'b'
  'i'
  't'