CSE115 / CSE503
Introduction to Computer Science I

Dr. Carl Alphonce

343 Davis Hall

alphonce@buffalo.edu

Office hours:

Tuesday 10:00 AM – 12:00 PM*

Wednesday 4:00 PM – 5:00 PM

Friday 11:00 AM – 12:00 PM

OR request appointment via e-mail

**Tuesday adjustments: 11:00 AM – 1:00 PM on 12/6*

ROADMAP

Last time

floating point representation

Today

wrapper classes

exercise 07

Coming up

exercises 08, 09 and 10

search (linear and binary)

ANNOUNCEMENTS

FINAL EXAM NOTICE

CHECK YOUR ENTIRE FINAL EXAM SCHEDULE!

http://blogs.advising.buffalo.edu/beadvised/posts/have-you-checked-your-final-exam-schedule-4/

Room assignments will be announced at a later date.

Lab 11

Due 9:00 PM on last day of classes for everyone.

This week – regular recitations and UTA office hours.

Next week – recitations are office hours.

Required functionality 80 pts.
Optional functionality – pick and choose.

100 points is full credit.
You can get more than 100 points.

Lecture exercises will give hints for lab 11 functionality.

Wrapper Classes

Java has eight primitive types

- boolean
- integral types:
  - signed: long, int, short, byte
  - unsigned: char
- floating point types: double, float

Type variables

- reference types OK
- primitive types NOT OK

Wrapper Classes

PRIMITIVE TYPE	WRAPPER CLASS
boolean	Boolean
byte	Byte
short	Short
int	Integer
long	Long
char	Character
float	Float
double	Double

Basic idea:

```
public class Integer {
    private int _value;
    public Integer(int v) {
        _value = v;
    }
    public int intValue( ) {
        return _value;
    }
}
```

Boxing, Unboxing, Autoboxing

```
Integer wi = new Integer(3);  // boxing
int i = wi.intValue( );       // unboxing

Integer wx = 3;               // autoboxing
int x = wx;                   // auto(un)boxing

int v = 2 * wi;               // autounboxing
Integer wv = 2 * wi;          // and boxing
```

EXERCISE 07

Define a method a method with this header in a class named quiz.Question:

```
public HashSet<HashSet<Integer>> answer(ArrayList<String> list)
```

Define the method so that it returns a partition* of list in which each subset consists of contiguous positions which have the same value in `list`.

Examples

null partitions into {}

[“”] partitions into {}

[“a”, “a”, “a”] partitions into {{0,1,2}}

[“a”,“b”,“b”] partitions into {{0},{1,2}}

[“a”,“a”,“b”,“b”,“c”,“c”,“a”,“a”,“a”] partitions into {{0,1},{2,3},{4,5},{6,7,8}}

* A partition of a set *X* is a set of nonempty subsets of *X* such that every element *x* in *X* is in exactly one of these subsets[2] (i.e., *X* is a disjoint union of the subsets).

https://en.wikipedia.org/wiki/Partition_of_a_set