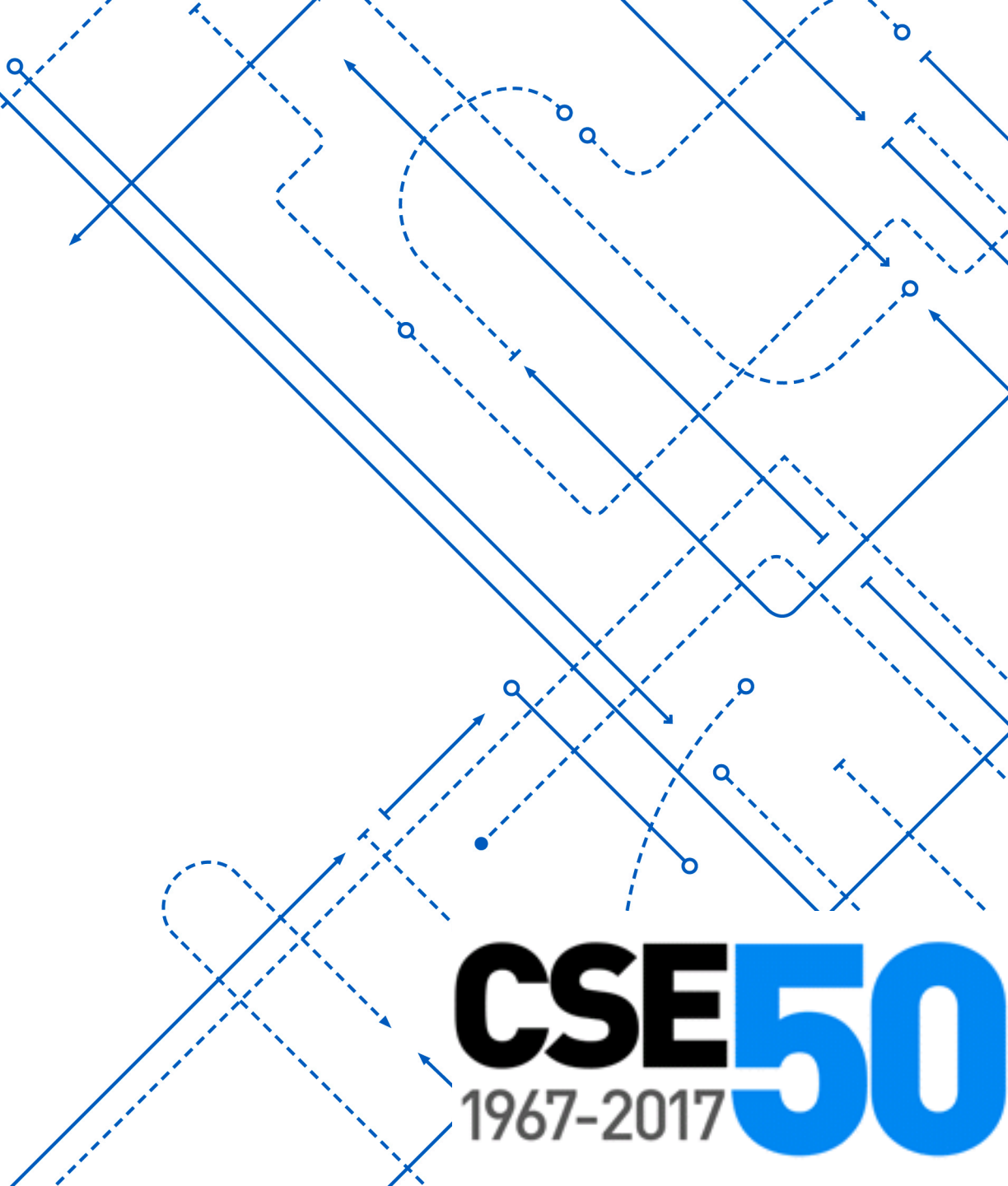
CSE 115 / 503
INTRODUCTION TO COMPUTER SCIENCE I

Dr. Carl Alphonce

Dr. Jesse Hartloff

University at Buffalo
School of Engineering and Applied Sciences



CSE 50
1967-2017

Last Day

CSE115 Introduction to Computer Science I

Announcements - 1

Lab 12 is due tonight at 8:00 PM for everyone.

HW 3 part 3 is due Monday morning at 8:00 AM for everyone.

Lab 12 and HW 3 hint slides are posted on the course website.

Review/Extra Help session – 7:00 PM to 9:00 PM: Monday 12/11 in Davis 113A

UTA OFFICE HOURS NEXT WEEK:
https://docs.google.com/a/buffalo.edu/spreadsheets/d/1Qek22cNWqJL1WDirqw3jIIRtFE0E4Q6YIusnOQ1PF9o/edit?usp=sharing

Announcements - 2

Finalized TopHat and Friday Activity points are posted in AutoLab.

The outstanding items are:

1) *Lab 12 (250 competency points, 2 proficiency points)*
2) *HW 3 part 3 (6 proficiency points)*
3) *Today's activity (250 competency points, 2 proficiency points)*
4) *Survey completion (2 proficiency points)*
5) *HW2 part 3 extra credit (2 proficiency points)*
6) *Final exam (1200 competency points, 10 proficiency points)*

You may know your score on the green items already.
The green and blue items will be up to date in AutoLab by Monday evening.
The orange item should be up-to-date by Tuesday evening.
The purple item will by up no later than the evening of Monday Dec 18.

Announcements - 3

Final exam room assignments will be posted on the course website AND will be e-mailed via UBLearns NEXT WEEK, prior to the exam.

You MUST attend your assigned room.
You MUST sit in your assigned seat.
You MUST bring your UBCard – it will be checked.

SURVEY

Take about 10 minutes now to complete the survey, if you have not already done so.

The link is in your e-mail, or you can type this:

https://tinyurl.com/ycl8qssc

PROBLEM 1

Explore what the following code does: trace the code by hand for various inputs and see what it produces as output. Then, come up with a plain English description of what the code does.

```
public ArrayList<String> mystery(ArrayList<String> a, ArrayList<String> b) {
    ArrayList<String> answer = new ArrayList<String>();
    int index = 0;
    while (index < a.size()) {
        answer.add(a.get(index));
        answer.add(b.get(index));
        index = index + 1;
    }
    return answer;
}
```

CHECKPOINT 1

What does the following code print?

```
ArrayList<String> list1 = new ArrayList<String>();
list1.add("a");
list1.add("b");
ArrayList<String> list2 = new ArrayList<String>();
list1.add("3");
list1.add("2");
System.out.println(mystery(list1,list2));
```

Recall the definition of mystery:

```
public ArrayList<String> mystery(ArrayList<String> a, ArrayList<String> b) {
    ArrayList<String> answer = new ArrayList<String>();
    int index = 0;
    while (index < a.size()) {
        answer.add(a.get(index));
        answer.add(b.get(index));
        index = index + 1;
    }
    return answer;
}
```

PROBLEM 2

Explore what the following code does: trace the code by hand for various inputs and see what it produces as output. Then, come up with a plain English description of what the code does.

```
public ArrayList<String> whatDoesItDo(ArrayList<String> a, ArrayList<String> b) {
    ArrayList<String> answer = new ArrayList<String>();
    int aIndex = 0; int bIndex = 0;
    while (aIndex < a.size() && bIndex < b.size()) {
        answer.add(a.get(aIndex)); aIndex = aIndex + 1;
        answer.add(b.get(bIndex)); bIndex = bIndex + 1;
    }
    while (aIndex < a.size()) {
        answer.add(a.get(aIndex)); aIndex = aIndex + 1;
    }
    while (bIndex < b.size()) {
        answer.add(b.get(bIndex)); bIndex = bIndex + 1;
    }
    return answer;
}
```

CHECKPOINT 2

What does the following code print?

```
ArrayList<String> list1 = new ArrayList<String>();
list1.add("k"); list1.add("v"); list1.add("w"); list1.add("b");
ArrayList<String> list2 = new ArrayList<String>();
list2.add("7"); list2.add("4");
System.out.println(whatDoesItDo(list1,list2));
```

Recall the definition of whatDoesItDo:

```
public ArrayList<String> whatDoesItDo(ArrayList<String> a, ArrayList<String> b) {
    ArrayList<String> answer = new ArrayList<String>();
    int aIndex = 0; int bIndex = 0;
    while (aIndex < a.size() && bIndex < b.size()) {
        answer.add(a.get(aIndex)); aIndex = aIndex + 1;
        answer.add(b.get(bIndex)); bIndex = bIndex + 1;
    }
    while (aIndex < a.size()) { answer.add(a.get(aIndex)); aIndex = aIndex + 1; }
    while (bIndex < b.size()) { answer.add(b.get(bIndex)); bIndex = bIndex + 1; }
    return answer;
}
```

See you at
the final!