CSE 121 – Lesson 9

Miya Natsuhara
Autumn 2023

Music: 121 23au Lecture Tunes 🐐

TAs: Trey Christina Sahej Vinay Kriti
    Sebastian Colton Anju Maria Minh
    Annie Janvi Jonus Shreya Vivian
    Jasmine Arkita Lydia Andy Nicole
    Christian Vidhi Luke Nicolas Simon
    Lucas Ritesh Andras Shayna Jessie
    Logan Hibbah Archit Hannah Lydia
    Jacob Julia Ayesha Aishah Yijia
Announcements, Reminders

• Creative Project 2 released – due Tuesday, Oct 31 🎃
• Resubmission Cycle 2 form released
  • Note: this is the last time C0 is eligible for resubmission.
• Quiz 1 on Thursday, Nov 2 in quiz section
• Mid-Quarter Formative Feedback with Ken Yasuhara for part of class on Wednesday, Nov 1
Common Problem-Solving Strategies

- **Analogy** – Is this similar to another problem you've seen?
- **Brainstorming** – Consider steps to solve problem before jumping into code
  - Try to do an example "by hand" → outline steps
- **Solve sub-problems** – Is there a smaller part of the problem to solve?
- **Debugging** – Does your solution behave correctly?
  - What is it doing?
  - What do you expect it to do?
  - What area of your code controls that part of the output?
- **Iterative Development** – Can we start by solving a different problem that is easier?
(PCM) Conditionals

if (test) {
    body (statements to be executed)
}

Executes a block of statements only if the test is true
(PCM) **Conditionals**

```java
if (test) {
  statement(s)
}
else {
  statement(s)
}
```

Executes a block of statements if the test is true, executes another block of statements if the test is false.
**Conditionals**

```java
if (test) {
    statement(s)
} else if (test) {
    statement(s)
}
```

Chooses between a block of statements to execute out of multiple choices, depending on which test it passes:
- If it ends in an else, exactly one block will be executed.
- If it ends in an else if, at most one block will be executed, but the code also may not execute any blocks of statements.
mEtAcOgNiTiOn iS ThE KeY To bEiNg sUcCeSSfuL In cOLLeGe
Common Problem-Solving Strategies

- **Analogy** – Is this similar to another problem you've seen?
- **Brainstorming** – Consider steps to solve problem before jumping into code
  - Try to do an example "by hand" → outline steps
- **Solve sub-problems** – Is there a smaller part of the problem to solve?
- **Debugging** – Does your solution behave correctly?
  - What is it doing?
  - What do you expect it to do?
  - What area of your code controls that part of the output?
- **Iterative Development** – Can we start by solving a different problem that is easier?
Common Problem-Solving Strategies

- **Analogy** – Is this similar to another problem you've seen?
- **Brainstorming** – Consider steps to solve problem before jumping into code
  - Try to do an example "by hand" → outline steps
- **Solve sub-problems** – Is there a smaller part of the problem to solve?
- **Debugging** – Does your solution behave correctly?
  - What is it doing?
  - What do you expect it to do?
  - What area of your code controls that part of the output?
- **Iterative Development** – Can we start by solving a different problem that is easier?
What is the output produced by executing this code?

```java
int a = 7;
int b = -1;
int c = 12;
if (a < b) {
    a *= 2;
} else if (b < a) {
    a /= 2;
} else {
    a = c;
}
if (c % 2 == 0) {
    c += 1;
}
if (b > 0) {
    b *= -1;
} else if (a < 0) {
    a *= -1;
}
System.out.println(a + " " + b + " " + c);
```

A. 7  -1  12
B. -3  -1  13
C. 3  -1  13
D. 12  1  12
E. -14  1  13