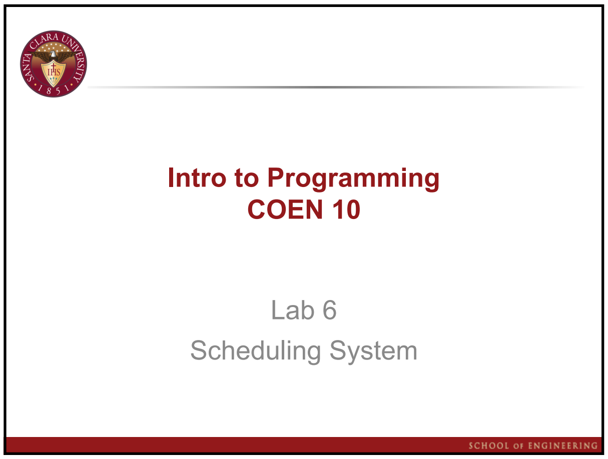
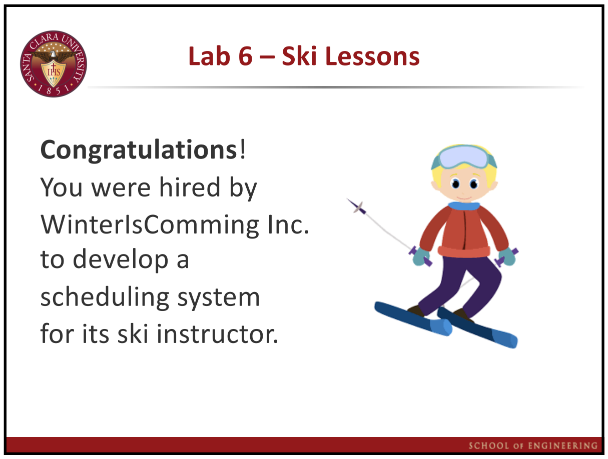


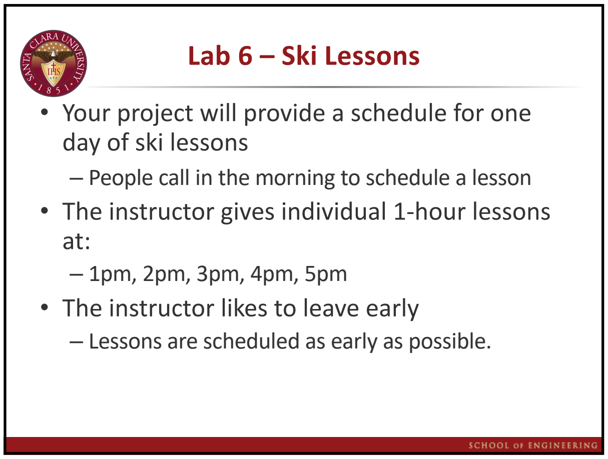
Intro to Programming
COEN 10

Lab 6
Scheduling System



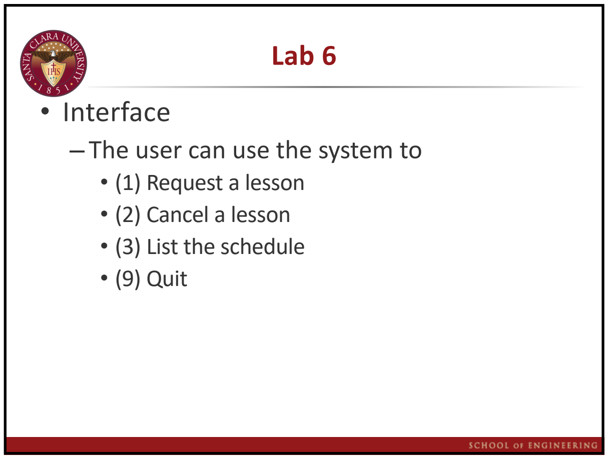
Lab 6 – Ski Lessons

Congratulations!
You were hired by
WinterIsComming Inc.
to develop a
scheduling system
for its ski instructor.



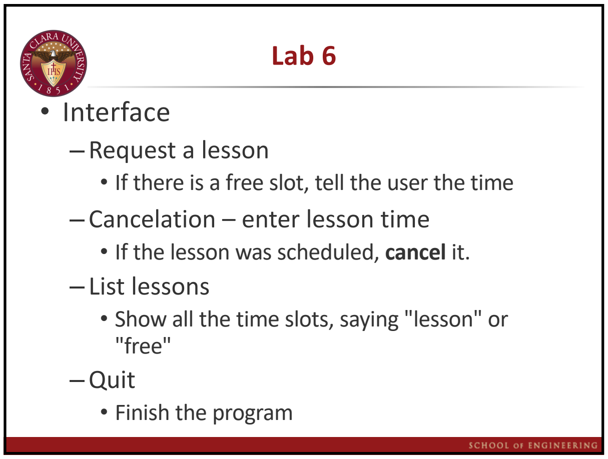
Lab 6 – Ski Lessons

- Your project will provide a schedule for one day of ski lessons
  - People call in the morning to schedule a lesson
- The instructor gives individual 1-hour lessons at:
  - 1pm, 2pm, 3pm, 4pm, 5pm
- The instructor likes to leave early
  - Lessons are scheduled as early as possible.



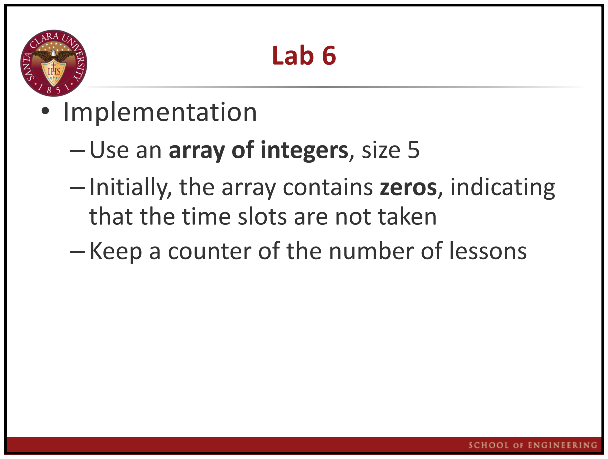
Lab 6

- Interface
  - The user can use the system to
    - (1) Request a lesson
    - (2) Cancel a lesson
    - (3) List the schedule
    - (9) Quit



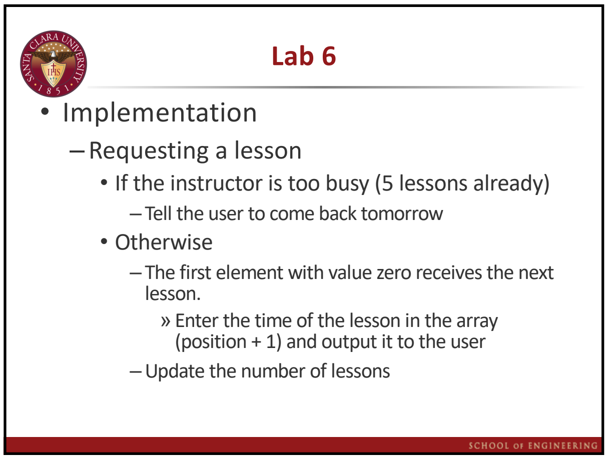
Lab 6

- Interface
  - Request a lesson
    - If there is a free slot, tell the user the time
  - Cancelation – enter lesson time
    - If the lesson was scheduled, **cancel** it.
  - List lessons
    - Show all the time slots, saying "lesson" or "free"
  - Quit
    - Finish the program



Lab 6

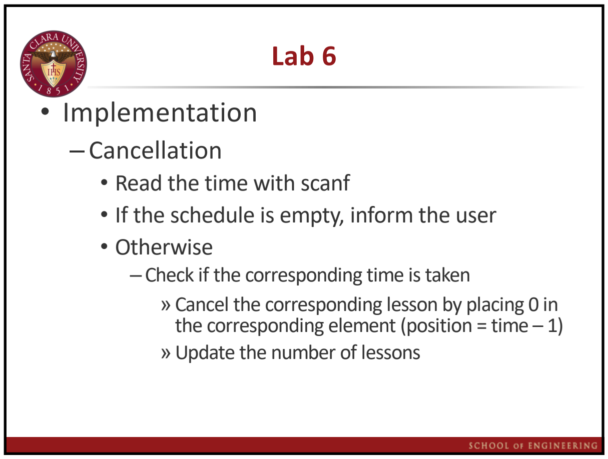
- Implementation
  - Use an **array of integers**, size 5
  - Initially, the array contains **zeros**, indicating that the time slots are not taken
  - Keep a counter of the number of lessons



Lab 6

- Implementation
  - Requesting a lesson
    - If the instructor is too busy (5 lessons already)
      - Tell the user to come back tomorrow
    - Otherwise
      - The first element with value zero receives the next lesson.
        - » Enter the time of the lesson in the array (position + 1) and output it to the user
      - Update the number of lessons

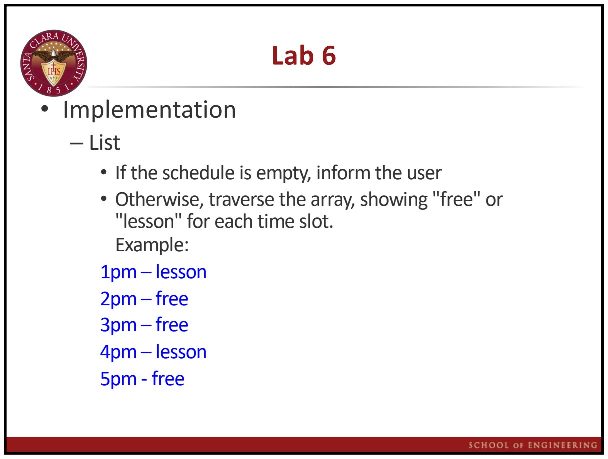
SCHOOL of ENGINEERING



Lab 6

- Implementation
  - Cancellation
    - Read the time with scanf
    - If the schedule is empty, inform the user
    - Otherwise
      - Check if the corresponding time is taken
        - » Cancel the corresponding lesson by placing 0 in the corresponding element (position = time – 1)
        - » Update the number of lessons

SCHOOL of ENGINEERING

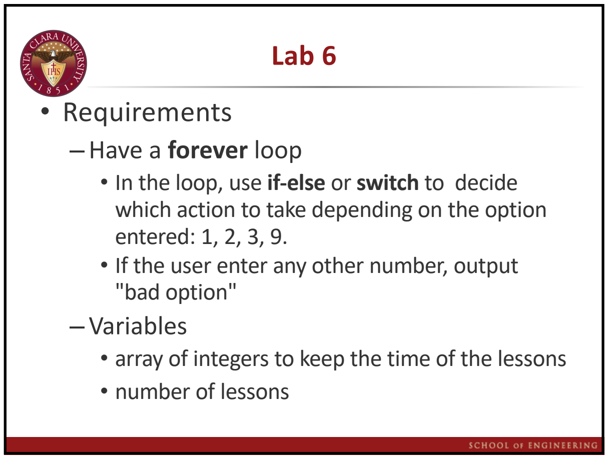


Lab 6

- Implementation
  - List
    - If the schedule is empty, inform the user
    - Otherwise, traverse the array, showing "free" or "lesson" for each time slot.
      Example:

```
1pm – lesson
2pm – free
3pm – free
4pm – lesson
5pm - free
```

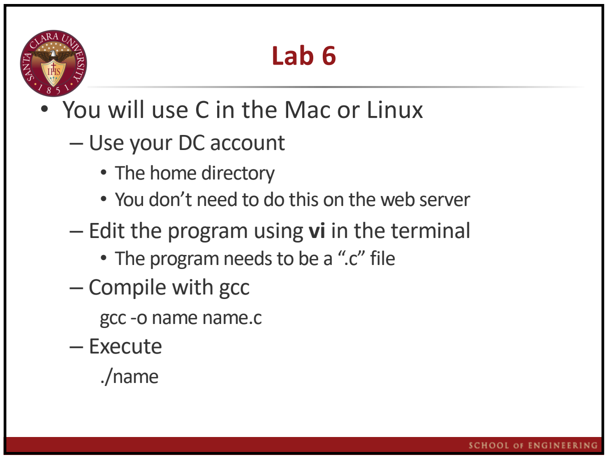
SCHOOL of ENGINEERING



Lab 6

- Requirements
  - Have a **forever** loop
    - In the loop, use **if-else** or **switch** to decide which action to take depending on the option entered: 1, 2, 3, 9.
    - If the user enter any other number, output "bad option"
  - Variables
    - array of integers to keep the time of the lessons
    - number of lessons

SCHOOL of ENGINEERING



Lab 6

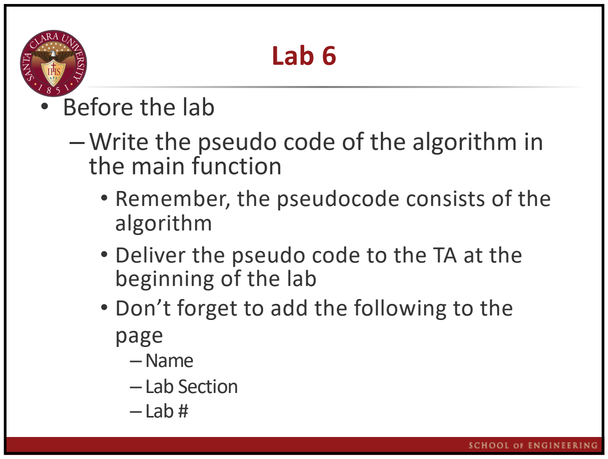
- You will use C in the Mac or Linux
  - Use your DC account
    - The home directory
    - You don’t need to do this on the web server
  - Edit the program using **vi** in the terminal
    - The program needs to be a “.c” file
  - Compile with gcc

```
gcc -o name name.c
```

  - Execute

```
./name
```

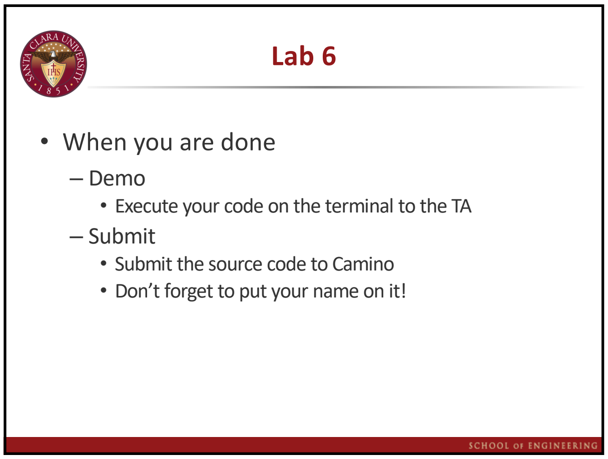
SCHOOL of ENGINEERING



Lab 6

- Before the lab
  - Write the pseudo code of the algorithm in the main function
    - Remember, the pseudocode consists of the algorithm
    - Deliver the pseudo code to the TA at the beginning of the lab
    - Don’t forget to add the following to the page
      - Name
      - Lab Section
      - Lab #

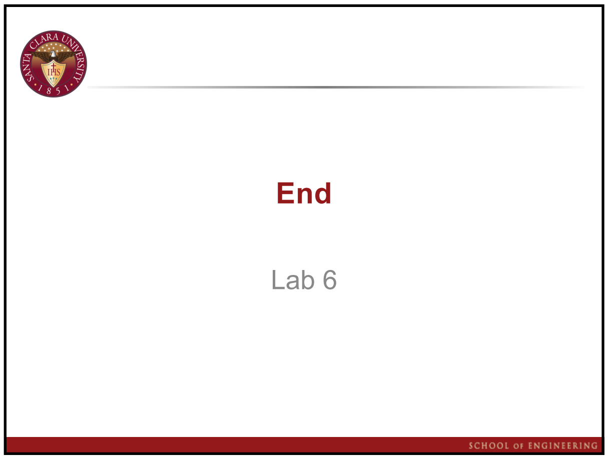
SCHOOL of ENGINEERING



Lab 6

- When you are done
  - Demo
    - Execute your code on the terminal to the TA
  - Submit
    - Submit the source code to Camino
    - Don't forget to put your name on it!

SCHOOL of ENGINEERING



End

Lab 6

SCHOOL of ENGINEERING