Intro to Programming
COEN 10

Lab 6
Scheduling System

Lab 6 – Pet Grooming Salon

Congratulations!
You were hired by StinkyPetSpa Inc. to develop a scheduling system for its store.

Lab 6

- Your project will provide a schedule for one day of grooming appointments
  - People call in the morning to schedule a session
- The groomer provides individual 1-hour grooming session at:
  - 1pm, 2pm, 3pm, 4pm, 5pm
- The groomer likes to leave early
  - Appointments are scheduled as early as possible.

Lab 6

- Interface
  - The user can use the system to
    - (1) Schedule an appointment
    - (2) Cancel an appointment
    - (3) List the schedule
    - (9) Quit

Lab 6

- Interface
  - Schedule an appointment
    - If there is a free slot, tell the customer the time
  - Cancelation – enter appointment time
    - If the appointment was scheduled, **cancel** it.
  - List schedule for the day
    - Show all the time slots, saying ”busy" 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 sessions

Lab 6

- Implementation
  - Scheduling a grooming session
    - If the groomer is too busy (5 appointments already)
      - Tell the customer to come back tomorrow
    - Otherwise
      - The first element with value zero receives the next session.
        - » Enter the time of the appointment in the array (position + 1) and output it to the customer
      - Update the number of appointments

Lab 6

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

Lab 6

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

1pm – busy

2pm – free

3pm – free

4pm – busy

5pm - free

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 sessions
    - number of sessions

Lab 6

- You will use C in the 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

Lab 6

- Before the lab
  - Write the pseudo code of the algorithm in the main function
    - Remember, the pseudocode consists of the algorithm
    - Show the pseudo code to the TA at the beginning of the lab

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!

End

Lab 6