

Lab Assignment Week 04

CSC 3320 – System-level Programming

Thursday, February 5th, 2026

Introduction

Welcome to the third programming lab of CSC 3320! Today, we will be covering the following topics:

1. C Fundamentals
 - Directives
 - Formatted I/O
 - C Strings

Lab Policies

- Attendance is mandatory.
- Labs must be completed **individually**.
- TAs are here to help you. Ask them for help!
- Lab assignments are due at midnight on the day of your lab.

Comments

The lab assignment requires the inclusion of comments to enhance code readability and understanding. Specifically, a block comment at the beginning of the C file is required. Your block comment should include the following:

- The program name
- The author's name (your name)
- A description of the program's overall purpose

Additionally, inline comments should be used throughout the code to explain specific lines or sections that might be less obvious to someone reading the code. These inline comments can clarify complex calculations, explain the purpose of certain variables, or provide additional context for specific code blocks.

Deliverables:

1. The C Code for your program. (.c file).
2. A screenshot of the output in the Terminal.

If you have any questions, please do not hesitate to ask your TA.

Program: Passenger Management System Lite

In this lab assignment you will implement the initial data entry functionality of an airline passenger management system. You will write a C program to read in information about a passenger and their flight information. Your program will use String Functions previously discussed in class to create a welcome message to the user with their flight information.

Program Requirements

- All variables must be of type `char[]`.
- Use macro directives to declare constants for all string lengths.

```
# define NAME_LEN 16
```

```
...
```

```
void main(void) {  
    char first_name[NAME_LEN];
```

- Prompt the user with a suitable message to read their first name, last name, airline, flight number, seat row number and seat letter

Enter the passenger's first name:

Enter the passenger's last name:

Enter the airline name:

Enter the flight number:

Enter the seat row number:

Enter the seat letter:

- Read the input of each prompt using `scanf()` into their own variables, respectively, with meaningful names.
 - Implement `scanf()` safely such that the user cannot exceed the maximum length of the string.
- Create a welcome message by combining user input with string literals using the string functions `strcpy()` and `strcat()`.
 - Utilize the "safer" `strncpy()` and `strncat()` so that the maximum length of the string cannot be exceeded.
- Display welcome message in the following format using `printf()`:

Welcome [First] [Last]! Your flight is [Airline] [Flight Number]. Your seat is [Seat Row][Seat Letter].

- You are not required to use Snowball to write this program. However, your program will be graded based on whether it works correctly on Snowball, so you should ensure it compiles and runs on Snowball.

Example Output

```
Enter the passenger's first name: Alan
Enter the passenger's last name: Turing
Enter the air line name: Delta
Enter the flight number: 5987
Enter the seat row number: 12
Enter the seat letter: C

Wlcome Alan Turing! Your flight is Delta 5987. Your seat is 12C
```

Deliverables

For today's lab, you must upload the C program code for your passenger management system and its output in the terminal on iCollege. Please name your C code and screenshot using the following format:

- C Files
 - lastname_firstname_filename.c
 - For example, **turing_alan_passenger.c**
- Screenshots
 - lastname_firstname_filename.png
 - For example, **turing_alan_passenger.png**