

COP 3223H: Introduction to C Programming

Fall 2023



University of
Central Florida

Dr. Kevin Moran

Week 1 - Class 1: Course Overview



Welcome to COP 3223H!



- Welcome to the First Lecture!



Instructor: Kevin Moran

Education: Ph.D. from William & Mary - 2018

Research Interests: Software Engineering ,
UI Analysis, Machine Learning

Office Hours: Mondays and Wednesdays, 12:00pm-1:00pm

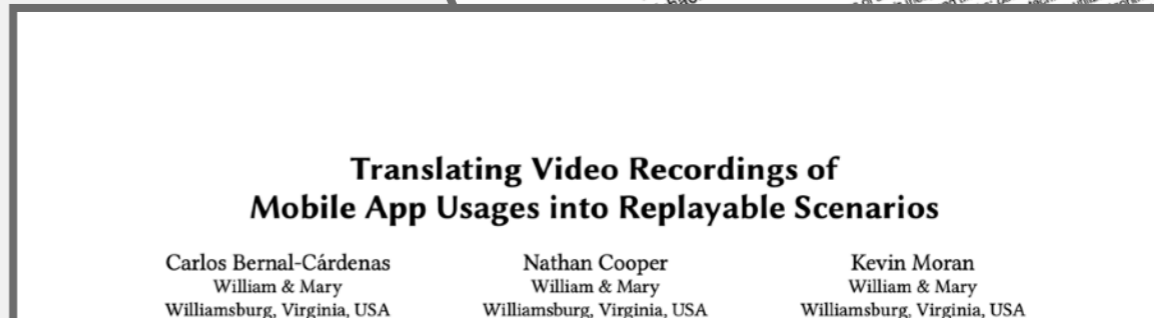


Instructor: Kevin Moran

Education: Ph.D. from William & Mary - 2018

Research Interests: Software Engineering ,
UI Analysis, Machine Learning

Office Hours: Mondays and Wednesdays, 12:00pm-1:00pm



spending development resources imple
After these initial design drafts are creat
they are faithfully translated into code
d-user to experience the design and user
ended form.
process (which often involves multiple
shown by past work and error prom
time-consuming, and error imp
of the design and imp
ms (which is
ally, UT

Introductions

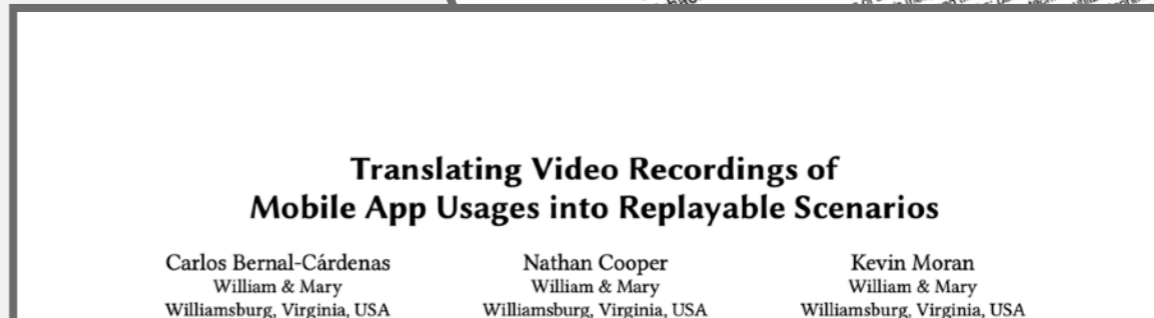
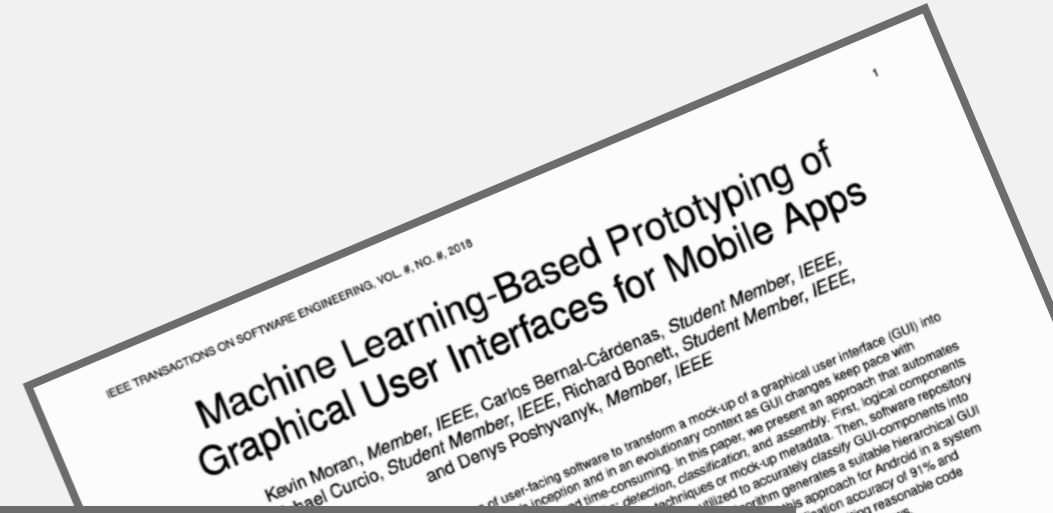


Instructor: Kevin Moran

Education: Ph.D. from William & Mary - 2018

Research Interests: Software Engineering, UI Analysis, Machine Learning

Office Hours: Mondays and Wednesdays, 12:00pm-1:00pm



Translating Video Recordings of Mobile App Usages into Replayable Scenarios

Carlos Bernal-Cárdenas
William & Mary
Williamsburg, Virginia, USA

Nathan Cooper
William & Mary
Williamsburg, Virginia, USA

Kevin Moran
William & Mary
Williamsburg, Virginia, USA

spending development resources imple
after these initial design drafts are creat
they are faithfully translated into code
d-user to experience the design and user
ended form.
process (which often involves multiple
shown by past work and error prom
time-consuming, and error imp
of the design and imp
ms (which is ally, UT

Recent Research








REPORTED STEPS

LAST THREE REPORTED STEPS

BURT HELP

Hi P90, please **select the app** that is having the problem

-  ATimeTracker v. 0.20
-  Android Token v. 2.10
-  AntennaPod v. 1.6.2.3
-  Droid Weight v. 1.5.4
- 

QUICK ACTIONS

- Finish reporting the bug
- Restart the conversation
- View the bug report

USEFUL TIPS

Feel free to **zoom in** this webpage on your browser to better see the chatbot and the app screenshots.

Write your message here ➤

Today's Agenda



- Provide an overview of the Course Logistics - (20-30 mins)
- Provide an Introduction to C and the C Compiler (10-15 mins)

Course Logistics





- I will not take attendance, however, you should come to class to learn and practice the concepts
- The presentation of topics follows a carefully planned sequence, as the comprehension of new material requires the mastery of topics previously presented.
- It is critical that students remain current in their study of topics, so as not to fall behind.



- Course Website: Syllabus, Schedule, Assignments, Lecture slides/recordings
- Webcourses (Canvas): Grades, Assignments, Discussions
- Zoom: Hybrid/Virtual Office Hours
- (Maybe) Piazza/Ed Discussions



Private | cs-ucf.github.io

COP 3223H - Fall 2023 - Introduction to C Programming

Home Schedule Assignments Syllabus Resources


Home

Course Description

Programming with C including arrays, pointer manipulation and use of standard C math and IO libraries.

General Course Information

Faculty



- **Instructor:** Dr. Kevin Moran
- **Office:** L3 Harris Engineering Center (HEC) 217A
- **Email:** kpmoran(at)ucf.edu
- **(Hybrid) Office Hours:** Monday 12:00am-1:00pm, Wednesday, 12:00pm-1:00pm or by appointment
-

[Join Office Hours](#)

Table of contents

- Course Description
- General Course Information
 - Course Meeting Times
 - Virtual Course Spaces
- Welcome Letter to Students
- Course Philosophy
- Learning Outcomes
- Course Grading Information
 - Grading Breakdown & Scale

Course Meeting Times



- There is no course textbook, however notes will be posted to the course website.
- There will be in-class activities for many lectures (*bring your laptop!*)

Grading Breakdown



- Exam 1 - (18%)
- Exam 2 - (18%)
- Final Exam - (18%)
- Large Programming Assignments - (20%)
- Small Programming Assignments - (16%)
- Quizzes - (4%)
- Eustis Setup - (3%)
- Syllabus Quiz - (1%)
- Entrance Survey - (1%)
- Exit Survey - (1%)



- Work together in small pairs/groups to gain experience trying out methods and concepts with examples
- No grades, but very important, as you will learn a lot from your classmates during these exercises



- Large Programming Assignments: Large programs will require the incorporation of multiple concepts into one programming assignment. Such programs solve a single problem, coding the solution in the C Language.
- Small Programming Assignments: Small programs consist of a set of problems solved by coding the solution in the C language. The solutions to the problems relate to specific concepts reviewed in the lecture(s).



- All assignments must be submitted through Webcourses
- All programming assignments can be turned in up to 24 hours late, but a 15% penalty will be applied
- Quizzes cannot be turned in late
- If you become ill, if you have a family emergency, or if something else happens that prevents you from completing work on time (either assignments or exams), please talk with me.



- Taken on Webcourses outside of class
- General concepts presented in the lectures with a mixture of multiple choice, true/false, and fill in the blank questions.
- All quizzes will have a time limit. Each quiz's time limit varies based on its content.
- You only get one chance for each quiz (they cannot be retaken).



- Writing code and answering questions about code
- You may be asked to determine if code contains bugs (both semantic and syntactic)
- There will be 3 exams, each worth 18%
- All exams are closed book
- See the Syllabus for more policies



- Learn the University and Course Policies for Academic Integrity
- HW Assignments are 100% individual
 - Discussing assignments at high level: **OK**, sharing code: **NOT OK**
 - If in doubt, ask the instructor
 - If you copy code, we **WILL** notice (see some of my recent research results on Code Traceability)



- OK Questions to ask your classmates:
 - *My code compiles with this strange error message. What does that mean?*
 - *Can you clarify what the instructor means in the directions for this assignment?*
 - *Can you explain what happens in memory when a variable is declared?*
 - *I am not sure how all these user defined functions in the large program assignment work together. Can you explain how they are all supposed to work together?*



- Questions that are NOT OK to ask your classmates:
 - *Can you show me how you implemented this user defined function?*
 - *How did you compare two arrays?*
 - *Is this how you check if the user input is valid and correct?*
 - *How did you take summation of all integer values stored in the 1D array?*
 - *Do you want to compare code solutions?*



- The use of AI tools to assist in coding are allowed in this class - but only in specific circumstances.
- Some things to remember:
 - Put Effort into Crafting High Quality Prompts
 - Be Aware of AI Limitations
 - Give the Tool Proper Attribution
 - Know when to use and not use AI Tools
- *More on this next class*



- My promises to you:
- Quiz results will be available immediately after the quiz is due
- Assignments will be graded within 1 week of submission
- Exams will be graded within 1 week

A Brief Introduction to C Programming



Why Learn How to Program?



- The world runs on code
- Coding is an incredible form of engineering, it allows to make abstract ideas concrete, and build incredible things.
- Programming allows us to solve a variety of problems
- It can help us to accomplish things that would otherwise be impossible through automation

Why Learn How to Program?



Why Learn How to Program?



150 Million Lines of Code

Some Basic Terminology



- Random Access Memory (RAM) - Stores Files and programs temporarily
- Central Processing Unit (CPU) -
 1. Fetch Instruction
 2. Execute Instruction
 3. Receives Next Instruction
- Instructions are in Machine Language
 - 1010111010110101

Programming Language Abstractions



- It would be difficult and tedious to write everything in machine code...
- Thus, we invented programming languages!
 - These are much closer to English and easier for humans to understand
 - However, they must be translated to machine code
 - This is the job of the compiler

The C Programming Language



- Dennis Ritchie of Bell Labs created the Programming Language in 1972
- C was the dominant language in the 1980s in the Unix ecosystem
- C was preferred for producing word processing programs, spreadsheets, compilers, and other products.
- C is easy to modify and easy to adapt to new models of computers.
- C remains the core language used for embedded systems

Why are we Learning C?



- The C language is a bit lower on the abstraction scale (closer to machine code)
- Because of this, you must reason about things that are abstracted by other languages (e.g., memory management)
- This will help you to think more critically about programming and develop a better mental model
- It will also help you to develop adept debugging skills!



- There are many different Text Editors that you can use, some of which provide assistive features for coding:
 - Notepad ++ (Windows)
 - Sublime Text (Windows + Mac)
 - Visual Studio Code (Windows + Mac)
 - Vim/Emas (Windows + Mac)

Our C Compiler (GCC)



- The compiler we will use to run our code in class is GCC
- Compiles Languages such as C, C++, Objective-C, Fortran, Ada, Go, and D
- Don't worry! You won't have to install the GCC compiler on your computer. You will have access to a Linux Server called Eustis that has GCC installed.



C Program Demo



ToDo List for this Week



- Complete Entrance Survey (Posted Tuesday)
- Test out a few different Text Editors for writing C code and pick one to use for the semester
- Make sure you can connect and run a C program on Eustis (Instructions will be posted soon).



Slides adapted from Dr. Andrew Steinberg's
COP 3223H course