# BASIC INFORMATION

## Course

| | |
|---|---|
| Name: | COMP 415 Mobile Computing First and For Most |
| Credits: | 4 |
| Prerequisites: | None |
| Term: | Fall 2016 |
| Class Meetings: | Thursday, 9 - 12 |
| Location: | P132 |
| Discovery Category: | Environment, Technology, and Society |

## Instructor


| | |
|---|---|
| Name: | Karen Jin, Ph.D., Assistant Professor |
| Program: | Computing Technology, Division of Science and Technology, UNH Manchester |
| Office: | P139 |
| Email: | karen.jin@unh.edu |


## Course Description

This course examines how mobile computing is transforming our everyday lives and the society and environment in which we live. In this course the students will engage the mobile ecosystem by inventing apps and solving problems of personal, social, and environmental relevance. Students will learn computational thinking skills and create mobile apps using App Inventor, a free and open source visual blocks-based programming environment. Students will share their creative apps with peers and communities. They will also exercise inclusion, civic engagement, and peer learning in the context of innovating with free and open source software that empower individuals and communities.
4 cr. ETS.

## Learning Objectives

Upon completion of this course students should be able to:

1. Create mobile apps using App Inventor programming environment and Android mobile devices.
2. Design user experiences and algorithms that solve problems of personal, social, and environmental relevance.
3. Explore and discover large datasets that affect diverse communities and the environment.
4. Communicate and collaborate in the creation of computational artifacts.
5. Connect computing with economic, social, and environmental contexts and with issues of diversity, inclusion, equity, and power.

## How to get in touch with me

There are three ways to get in touch with me:

1. **Post questions on class forum**.

2. For in-person communication, see me **after class** or **by appointment**. Appointments can be made through http://calendly.com/karenajin You may meet with me online with Skype/Hangout/Zoom during my office hours.
3. You may also email me at karen.jin@unh.edu for questions or arrange an individual meeting outside my office hours.

## Learning Resources

### Textbook

***App Inventor 2: Create Your Own Android Apps***, by David Wolber, Hal Abelson, Ellen Spertus, and Liz Looney. O'Reilly Media, Inc., Oct 2014, 2nd edition. Also available through the **UNH Library, Safari Books Online** digital library. Web page draft versions of the AI2 book are available at http://www.appinventor.org/book2.

### Phones

You will be loaned an unlocked Android phone with a memory card (secure digital SD) for the duration of the semester. You are expected to return at the time you take the final exam. If you lose the device, you'll be charged $100. If you don't pay this balance by the end of the semester, final grade submission will be on hold until the payment is done. You may use your own Android phones, if you have one. The phone must have an SD card to store your apps.

### Development Platform, Resources, and Tools

- **Platform: MIT App Inventor Version 2** at http://ai2.appinventor.mit.edu is a web-based development environment maintained by the MIT's Center for Mobile Learning @ MIT Media Lab. The App Inventor development environment lets you build apps for Android devices using a browser on your computer. To have access to the App Inventor development environment you need to have a Google Account.
- **Resources**
    - MIT App Inventor at http://appinventor.mit.edu/explore/
    - David Wolber's **appinvetor.org - app building for everyone** site at http://www.appinventor.org/.
- **Tools**: free and open source tools will be used to create multimedia resources for the apps, such as pictures and graphics, audio, and video resources. There are also free mobile apps that you need to install on your development phone to do "live testing" of your app, QR code reading, and more.

### Tech Consultants

Computing Technology department has tech consultants who are available to help with software configuration and other technical questions you might have.

## Instructional Approach

"*For the things we have to learn before we can do them, we learn by doing them*", Aristotle, ~350 BC

Learning in this class depends heavily on *active participation* and *open collaboration* in and outside class. The course has 15 weeks with Wednesday 2:50 hour class meetings. You are expected to study 6-8 hours outside class every week.

Learning activities are structured by blending *in-person* and *online time* to engage with the course content:

- During class time we participate in discussions, presentations, solution review, and guided lab

activities.
- Outside class time is for studying concepts and techniques, applying them to solve problems, doing homework , giving feedback to peers, reflecting on one's own work, collaborating with peers, and working on team projects.
- Online means of communication and collaboration include the class forum, learning resources repository, and learning portfolios.
- In-class collaboration uses pair programming and group deliberations for designing, coding, and discussion.

# GOALS and LEARNING OBJECTIVES

## Course Goals

Mobile computing is ubiquitous and fast growing. The course is an introduction to mobile computing and a learning opportunity to experience computational thinking – a way of thinking that benefits everyone, not only for computing professionals. We care about computational thinking because it engages us more fully in the modern data-rich and interconnected digital world.

Computational thinkers express problems as computational threads that can be woven in solutions that computing devices carry out. Practices of computational thinking consist of:
- Recognizing and defining computational problems
- Developing and using abstractions
- Creating computational artifacts
- Testing and iteratively refining solutions.

Computational thinking practices above encompass solving computational problems and creating computational artifacts. They overlap and are integrated with:
- Fostering an inclusive and diverse computing culture
- Collaborating on computational tasks with other individuals and teams
- Communicating about computing.

Computational thinking practices are developed by exploring big ideas in computing:
- ***Creativity***: embraces creative expression the exploration of ideas to create prototypes and solve computational problems
- ***Abstraction***: extract common features from examples, create more general modes, reuse in different situations
- ***Data analysis***: collect, store, transform for communication, and make inferences based on data
- ***Algorithms***: design computational steps and threads to accomplish specific tasks
- ***Programming***: computational principles, processes, and tools to develop programs
- ***Computing systems, networks, and the Internet***
- ***Impacts of computing*** on culture, social interactions, law, and ethics

The course seeks to broaden participation in computing from students with llong-standing underrepresentation in computing: women, persons with disabilities, underrepresented students of color, and underserved students, such as low-income and first generation students.

## Attitudes and Dispositions

*"Success is going from failure to failure without loss of enthusiasm*", Winston Churchill
Achieving the course learning objectives will, hopefully, form the following attitudes and dispositions:
- Tolerate uncertainty and ambiguity
- Persist in working with difficult problems

- Adapt, adjust, change course, be flexible as needed
- "Walk in another's shoes" to learn more about other perspectives
- Have confidence in dealing with complexity
- Communicate and work with others to achieve a common goal.

# COURSE REQUIREMENTS

## Learning Portfolios and Practice

All your work is uploaded to your student learning portfolio. You will use the Google Drive to create and maintain your portfolio. Portfolio work includes: labs, homeworks, reflections, and the team project. Portfolios will be evaluated weekly during the semester.

## Homework Assignments

There will be **five** outside class homework assignments, each **4 points**, for a total of **20 points** of the final grade. Homework assignments must be completed in your portfolios before the class when when they are due (no later than midnight, 11:59 PM, the day before the class). Homework assignments require completion of assigned readings, programming practice, solving problems, Python coding, asking questions, and giving feedback.

## Learning Reflections

Reflecting on your learning experience is facilitated by **weekly reflections** you write in your portfolio. There are **12** required reflections (and 2 optional) for a total of **12 points**. Weekly reflections must be completed and uploaded to your portfolios before class (no later than midnight, 11:59 PM, the day before the class). Reflections are well-thought answers to guiding questions. No credit is received for big, general statements that say little about your specific experience. To receive credit you must describe very specific instances, whether a challenge, an achievement, or collaboration situation. Descriptions must be followed by explanations.

## Collaboration and Lab Practice

You are required to collaborate **in class** with your partner using **pair programming**, with your peers through **discussions** and **demonstrations**, and with your team members while working on the project. Practice outside class is facilitated by the class forum on my courses. Class forum contributions are questions, answers, follow-ups, or edits to other contributions.

Class time always requires lab practice: getting starting on app development projects that you might complete outside class. Evidences of lab practice are app development artifacts in your portfolio (**8 points**).

## Creative Projects

There will be a team project to create a mobile app that has to be proposed (pitched), designed, implemented, and presented to a general audience. The creative project will address an issue of interest to the team.

Project artifacts include two project logs (**6 points**), codebase (**5 points**), ***Winter Science Symposium*** poster (**6 points**) to be presented on ***Wednesday December 7*****, 3 - 5 pm**, and final report (**3 points**), for a total of **20 points**. All project artifacts must be completed and uploaded to your portfolios before class (no later than midnight, 11:59 PM, the day before class). Timely communication and open collaboration with your team members factor into the grading of your project work.

### Exams

There will be a **midterm** (7th week of the semester) and comprehensive **final exam** (last week of the semester). Each counts **20 point**s to the final grade, for a total of **40 points**. The exams are closed books, paper and pencil based, and assess problem solving skills, application of concepts, and demonstration of programming techniques.

## GRADING AND EVALUATION OF STUDENT WORK

To learn in this class you do homework assignments, work on the team creative project, do all your work in the learning portfolio, take a midterm and final exam, contribute to the class forum, and reflect on your weekly progress.

Final grade is calculated as follows:

- **5 homework assignments** @ **4 points** each, for a total of **20 points**
- **12 weekly reflections @ 1 point** each, for a total of **12 points**
- **Lab practice**, **8 points**
- **Team project, 20 points**, broken down into:
    - **2 logs @ 3 points** each, for a total of **6 points**
    - **codebase, 5 points**
    - **Winter Science Symposium poster & presentation, 6 points**
    - **final report, 3 points**
- **Midterm exam** and **final exam @ 20** points each, for a total of **40 points**.

There is a **5 points** penalty for each unexcused absences (see policy on **Attendance** below).

## COURSE POLICIES REGARDING STUDENT BEHAVIOR

### Late submissions and make-up exams

You will lose 10% per day for work that are submitted late. Late work is any homework assignments and project submitted after the announced due date and time. Three days after the due date late work will not be accepted and a zero will be given unless you have received prior approval to submit after this time.

Make-up exams will be given only in case of a serious emergency. You must show evidence that you are unable to take the exam, such as a doctor's note. No make-ups will be granted for personal reasons such as travel, personal hardship or leisure.

### Student use of computing devices

In-class use of any computing device is **not allowed** unless needed for lab activities and with the instructor's permission. Use of computing devices for non-class activities is not allowed. You will be asked to leave the class if you fail to comply with these rules. Students with a learning disability that requires the use of a computing device must provide evidence from the Disabilities Services office.

### Academic honesty

No collaboration is allowed while taking the exams. Cheating on the exam is penalized with failing the course.

Assignment submissions should be entirely your work and may not include work done by others. Collaboration on assignments is encouraged, but does not include preparing and submitting the final artifacts that are uploaded to your portfolio.

Failing to comply with these rules is considered a violation of academic honesty policy.
See http://www.unh.edu/student-life/handbook/academic-honesty for more information. There are very serious repercussions if you deviate from the academic honesty policy:

- The penalty for the first occurrence of an instance of academic dishonesty and plagiarism is no credit for the assignment in question. The Associate Dean will be immediately notified of the incident.
- The second attempt is penalized with failing the course.

## STUDENTS WITH DISABILITIES

UNH Manchester is committed to providing students with disabilities with a learning experience which assures them of equal access to all programs and facilities of the University, which makes all reasonable academic aids and adjustments for their disabilities and provides them with maximum independence and the full range of participation in all areas of life at UNH Manchester. Students who need to document their disability and determine any accommodations, services, or referrals should schedule an appointment with the UNH Manchester Disability Services Coordinator by calling 641-4170. For more information, please seehttp://manchester.unh.edu/disability-services.

## TENTATIVE COURSE SCHEDULE

---

**Lectures:** (**#sp** or **s**lide **p**resentation), **Homeworks** (**H#**)
**Reading**: Wolber et al. (**W#**)
**Project**: Project logs (**PL#**), poster, project presentation, codebase, final project report.

| Wk # | Date | Core computational concepts, lab experience, and computational thinking practices | Due next week | |
|---|---|---|---|---|
| | | | Reading | Homework, Reflection, Project |
| 1 | 9/1 | Exploring coding and computational thinking. Mobile app development language and tools. Event-driven computation. Learning portfolios, forum, and pair programming. Lab: **hellomassabesic** | W1, W14 | R1 |
| 2 | 9/8 | Event-driven design. Animations with canvas, sprites, and clock components. Built-in data types: numerical, text, and boolean. Remembering data with variables. Making decisions with conditionals. Lab: **cleanmassabesic** | W2, W3; partial W16, W17, W18 | H1, R2 |
| 3 | 9/15 | List data type, values, and operations. Repeating actions with loops. Interactions with apps (by the user or other systems): text to speech, texting, Google Voice. Persistent data. Lab: **notextingwhiledriving** | W4, W23; partial W19, W20, W22 | R3 |
| 4 | 9/22 | Location sensing and other mobile computing features. Computational expressions and functions. More on list processing. Procedural abstractions. Lab: **parktour** | W6, W7, W19; partial W21, W22, W23 | H2, R4 |
| 5 | 9/29 | More on list iteration, conditionals, and text processing. Project ideas, teams, and timeline. Lab: **bugcollection** | W18, W19 | PL1, R5 |
| 6 | 10/6 | ***Midterm review*** | | R6 |

| | | | | |
|---|---|---|---|---|
| **7** | **10/13** | **Midterm exam** | | R7 |
| 8 | 10/20 | More general and reusable procedures. Engineering and debugging apps. Drafting project proposal | W15, W21 | H3, R8 |
| 9 | 10/27 | More on sensors. Datasets and data stores. Local persistent data. Project work | W22, W23 | PL2, R9 |
| 10 | 11/3 | Launching other apps with the ActivityStarter component. Other user interaction components. | | H4, R10 |
| 11 | 11/10 | Shared data. Data store services. Project work | | H5, R11 |
| 12 | 11/17 | Project work. Winter Science Symposium poster. Project work | | R12, project poster |
| ***Thursday, November 24, No Classes. Enjoy Thanksgiving Holiday*** | | | | |
| 13 | 12/1 | More project work. | | R13, project report |
| **14** | **12/7** | ***Final exam review.*** **Winter Science Symposium poster presentations 3 - 5 pm** | | R14 |
| **15** | | **Final exam** | | |