COURSE CONTENT

Course Code	DM2008
Course Title	Programming for Interaction
Pre-requisites	NIL
No of AUs	3
Contact Hours	39 hours studio contact

Course Aims

This is an introductory course that will introduce you to the fundamental concepts shared by all core programming languages. The core structure and foundations of common programming languages will be explored in a theoretical and practical manner as well as demonstrating that sometimes vastly different applications such as games, websites and installations are based on similar logical and structural principles. By the end of the course, you will be able to write basic games, and create interactive environments. This will allow you to create digital systems and artwork individually or collectively in future practice.

Intended Learning Outcomes (ILO)

By the end of the course, you should be able to:

1. Produce animated 2D and 3D graphics using a code authoring environment.
2. Write code which accepts input from devices to drive interactions in environments using software or micro controllers.
3. Apply coding tutorials and exercises to the creation of interactive media projects.
4. Present, discuss and evaluate coding projects in a clear and cohesive manner.
5. Contribute to the learning environment by participating positively to class discussion, critiques and activities related to coding and interactive media.

Course Content

Statements, Functions, Variables, and Arrays
Overview of Control statements (while, for); Conditional Statements (if, else if, else, switch); Functions (Input arguments and return values; Variable types and casting; and Arrays (1-d, 2-d)

Objects and Libraries
An introduction to objects, object constructors, overloaded constructors, overloaded functions; the importance of and using libraries; important statements; and object subclassing and inheritance.

Introduction to Arduino
An exploration of Arduino; what is an Arduino? Why would you want to use one? Example uses of an Arduino; LED's, PWD, and voltage dividers.

3D Graphics and Processing
An introduction to the 3D graphics environment; demonstrations of rotations on different axes; light types and directions; data structures (Queue, Stack); drawing techniques, drawing in 3D spaces; coordinate system fundamentals; texture mapping.

Python
An introduction to Python; typeless languages; syntax differences from Java (for, while, if, etc.); introduction to the IDE; useful Python libraries; Python pip.

Class assignments
Nine programming tutorials, exercises that explore the application of programming for interaction design; mid-term project to create an Arduino device; a final project that demonstrates sufficiency in programming. Developed through lectures, tutorials, class exercises and peer/instructor feedback sessions.

Assessment (includes both continuous and summative assessment)

Component	ILO Tested	Programme LO	Weighting	Team/ Individual
Continuous Assessment Tutorials In-class exercises Arduino project Assignment presentations	1,2,3,4	--	40	Individual
Final Project: Each student develops a creative final project that incorporates programming skills learned in the course.	1,2,3,4	--	40	Individual
Continuous Assessment: Participation	5	--	20	Individual
Total			100%	

Reading and References

1. Igoe, Tom. *Making Things Talk: Using Sensors, Networks, and Arduino to see, hear, and feel your world* . O'Reilly 2011
2. Shiffman, Dan. *Learning Processing, Second Edition: A Beginner's Guide to Programming Images, Animation, and Interaction* Morgan Kaufmann. 2015.
3. Watanabe, Ken. *Problem solving 101: A simple book for smart people*. Penguin, 2009.

Course Policies and Student Responsibilities

(1) General

You are expected to complete all assigned readings, activities, assignments, attend all classes punctually and complete all scheduled assignments by due dates. You are expected to take responsibility to follow up with assignments and course related announcements. You are expected to participate in all project critiques, class discussions and activities.

(2) Punctuality

You are expected to be punctual for all classes. If you are more than 30 minutes late, you will be deemed as absent and will not be able to sign on the attendance register.

(3) Absenteeism

In-class activities make up a significant portion of your course grade. Absence from class without a valid reason will affect your participation grade. Valid reasons include falling sick supported by a medical certificate and participation in NTU's approved activities supported by an excuse letter from the relevant bodies. There will be no make-up opportunities for in-class activities.

Academic Integrity

Good academic work depends on honesty and ethical behaviour. The quality of your work as a student relies on adhering to the principles of academic integrity and to the NTU Honour Code, a set of values shared by the whole university community. Truth, Trust and Justice are at the core of NTU's shared values.

As a student, it is important that you recognize your responsibilities in understanding and applying the principles of academic integrity in all the work you do at NTU. Not knowing what is involved in maintaining academic integrity does not excuse academic dishonesty. You need to actively equip yourself with strategies to avoid all forms of academic dishonesty, including plagiarism, academic fraud, collusion and cheating. If you are uncertain of the definitions of any of these terms, you should go to the academic integrity website for more information. Consult your instructor(s) if you need any clarification about the requirements of academic integrity in the course.

Planned Weekly Schedule*

*Subject to adjustment by instructor according to the teaching situation, students' progress, public holidays and unforeseeable circumstances. A revised schedule will be issued to students at the start of the semester.

Week	Topic	Course LO	Readings/ Activities
1	• **Course Overview** Introduction to the instructor, grading, syllabus, why we are using the languages of Java and Python; description of environments used (Processing, Arduino IDE, Thorny.	1,2, 3, 4, 5	**Introductory Lecture** **In-class discussion** on the techniques of programming for interaction design. **Assigned Tutorial:** Introduction to Processing
2-3	• **Statements, Functions, Variables, and Arrays** Overview of Control statements (while, for); Conditional Statements (if, else if, else, switch); Functions (Input arguments and return values; Variable types and casting; and Arrays (1-d, 2-d)	1,2, 3, 4, 5	**Lectures on Statements, Functions, Variables and Arrays** **Presentation of the Prcoessing Tutorial.** Critique and feedback. **Assigned Tutorial:** Building Pong Part I and II
4-5	• **Objects and Libraries** An introduction to objects, object constructors, overloaded constructors, overloaded functions; the importance of and using libraries; important statements; and object subclassing and inheritance. -	1, 2, 3, 4, 5	**Lectures on Objects and Libraries** **Presentation of the Building Pong Tutorial.** Critique and feedback. **Assigned Tutorial:** Flappy Bird Part I and II **Additional Homework:** Programming review questions

6-7	• **Introduction to Arduino**
	An exploration of Arduino; what is an Arduino? Why would you want to use one? Example uses of an Arduino; LED's, PWD, and voltage dividers.
	  -
	  -
		1, 2, 3, 4, 5	- **Lecture on Arduino**
			I**n-class Work** mid-term project to build an Arduino device.
			Mid-term Project Presentation
			Each student will present their Arduino project for critique and feedback.

8, 9	• **3D Graphics and Processing** An introduction to the 3D graphics environment; demonstrations of rotations on different axes; light types and directions; data structures (Queue, Stack); drawing techniques, drawing in 3D spaces; coordinate system fundamentals; texture mapping.	1, 2, 3, 4, 5	- **Lecture on 3D Graphics and Processing** **Assigned Tutorial 1:** Arduino Lab 1- various LEDS **Assigned Tutorial 2:** Arduino Lab 2- Sensors, Motors, and Servos **Presentation of Arduino Lab 1** Critique and feedback. -

10, 11	- **Python** An introduction to Python; typeless languages; syntax differences from Java (for, while, if, etc.); introduction to the IDE; useful Python libraries; Python pip.	1, 2, 3, 4, 5	- **Lecture on Python** **Presentation of Arduino Lab 2** Critique and feedback. **Assigned Tutorial:** Python coding questions I**n-class Work** final project

12	- **Course Review**
	  -
	  - Summary of programming languages and technical skills learned through the course to be applied to the final project.
			- **Lecture on Course Review**
			In-class Discussion: each students gives a brief overview of their final project and how it incorporates skills, techniques, and concepts learned in the class.
			In-class Work final project
			  -

13	- **Final Project Presentations**		**Final Project Presentation** - Each student will present their final project for critique and feedback.