

Miyazaki International College
Course Syllabus
Fall 2022

Course Title (Credits)	IDS313 - Algorithmic Thought (3 credits)
Course Designation for TC	N/A
Content Teacher	
Instructor	Anderson Passos, Ph.D.
E-mail address	apassos@sky.miyazaki-mic.ac.jp
Office/Ext	学部長室
Office hours	月曜－金曜 14:00-16:00

About this instructor:

Anderson has graduated from Computer Science and worked in the IT industry for over 14 years, including 4 years of experience as System Engineer in Tokyo, Japan.

パッソス先生はコンピュータサイエンスを卒業し、14 年以上にわたり IT 業界で働きました。これには、日本の東京でシステムエンジニアとして 4 年間の経験があります。

Course Description:

Critical thinking is a requirement in everything we do. Being able to evaluate the surroundings (the context) and come up with a solution will dictate students' development personally and professionally. Students will learn how to organize their ideas and to connect their thoughts through the development and resolution of algorithms. Students will also have the chance to experiment with real algorithm development through the application of basic programming language skills. This course is highly recommended for all students planning to take Fundamentals of Computer Programming.

Course Goals/Objectives:

This course will provide students with tools for developing a critical sense of problem analysis and organization of ideas. They will often be challenged with algorithms that will gradually increase in difficulty level as students' problem solving abilities evolve. Students will learn about:

- Problem interpretation
- Problem/Solution evaluation
- Procedural thinking
- Critical thinking
- Practice algorithmic problem solving with minimal aid of computers, emphasizing paper-based strategies.
- Basic contact with computer programming

Teaching Methodology:

Course objectives will be achieved through a variety of active learning teaching strategies, including but not limited to:

Active Learning Teaching Strategy	Course Schedule
Interactive lectures with note taking	Every class
Pause for Reflection	Every class
Pair & Group work	Every class
Simulation and Experiments	Final Project development

Tentative Course Schedule

Day	Topic	Content/Activities
1	Introduction	<ul style="list-style-type: none"> - Syllabus review - Introduction to course topics
2		<ul style="list-style-type: none"> - Multiple solution problems - Activity: People attending an event
3		<ul style="list-style-type: none"> - How do an algorithm work - Black box concept
4	Algorithm Structure	<ul style="list-style-type: none"> - Structured language
5		<ul style="list-style-type: none"> - Understanding algorithms - Understanding programs
6		<ul style="list-style-type: none"> - Computer model - Address vs data - Basic computer instructions
7		<ul style="list-style-type: none"> - Instruction execution - Program execution
8		<ul style="list-style-type: none"> - Introducing abstraction <ul style="list-style-type: none"> o Instructions o repetitions
9	Elementary Concepts	<ul style="list-style-type: none"> - Algorithms and programming languages
10		<ul style="list-style-type: none"> - Problem breakdown - Activity: Calculating roots of a second degree equation

11	Elementary Techniques	<ul style="list-style-type: none"> - Variables inside repetitions - Conditional statements
12		<ul style="list-style-type: none"> - Repetitions inside conditions - Conditions inside repetitions - Activity: Checking if a number is odd or even
13	Technical Applications	<ul style="list-style-type: none"> - Fibonacci - Factorial - Series - Prime numbers
14	Functions and Procedures	<ul style="list-style-type: none"> - Functions vs Procedures
15		<ul style="list-style-type: none"> - Modularity - Code reuse <ul style="list-style-type: none"> o Legibility o Adding comments
16		<ul style="list-style-type: none"> - Main program - Global variables - Local variables <ul style="list-style-type: none"> o
17		<ul style="list-style-type: none"> - Using functions and procedures <ul style="list-style-type: none"> o Changing variables inside functions o Parameters by value o Parameter by reference
18	Data structures	<ul style="list-style-type: none"> - Complex data types - Data access
19		<ul style="list-style-type: none"> - Linked data structures - Persistent data structures
20		<ul style="list-style-type: none"> - Arrays - Associative arrays
21		<ul style="list-style-type: none"> - Graphs - Matrix
22	Sorting algorithms	<ul style="list-style-type: none"> - Algorithm design - Algorithm comparison

23		- Insert - Selection
24		- Merge - Heapsort - Quicksort
25		- Bubble - Shell - Comb
26		- Counting - Bucket - Radix
27	Computer Programming	- Program design - Program Documentation
28		- Software Design Specification - Software Development
29		- Software flow - User interface and accessibility
30		- Content review for finals
	Finals	

Course Policies (Attendance, etc.)

Student Responsibilities

As a class member, you are responsible for attending all classes and arriving on time, for participating as a member of a group, and for completing and handing in all assigned work.

Attendance and Lateness

- You can be absent a maximum of 4 times
- If you are more than 20 minutes late, you will be given an absence
- Two lateness are equal to one absence

If you sum up more than 4 absences (e.g. 3 absences and 3 lateness), you will be asked to withdraw from the course. Failing to do so will result in an automatic "F" grade.

Homework

The homework should be handed in at the beginning of every class. All homework submitted

after that will **not** be considered.

Excused Absences

Excused Absence forms should be brought to instructors as soon as possible. If you know you will be missing classes talk to us beforehand so we can arrange make-up work.

Late Assignments

It is your responsibility to look for the instructor and check what assignments are due. Also, if you miss a class, it is your responsibility to contact the instructors to get handouts and explanations (missing a class is no excuse for not handing in homework).

Plagiarism and Intellectual Honesty

Plagiarism is representing someone else's intellectual property--words, ideas, or images-as your own. It is a very serious academic offense and plagiarized work is not accepted in this course. Ask one of the instructors if you have any questions about this. You are responsible for understanding what plagiarism is and knowing how to avoid it in your work.

Required Materials:

- notepad

Class Preparation and Review

Students are expected to spend at least one hour preparing for every hour of lesson, and one hour reviewing and doing homework. Make sure you review your notes after each class and make sure you understand the topics covered. Instructors are available outside the classroom in case students need additional assistance (please check office hours on the first page of this syllabus).

Very often, you will have homework assignments. Make sure you review those before coming to the next class. Also, check the class Content/Activities so you can be prepared for the class.

Grades and Grading

- | | |
|-----------------|-----|
| • Homework | 10% |
| • Projects | 50% |
| • Tests | 20% |
| • Participation | 20% |

You will receive a mid-semester evaluation to let you know how you have been doing up to that

point.

Notes:

Course Rubrics

ELEMENT	Exemplary (3 points)	Proficient (2 points)	Partially Proficient (1 point)	Unsatisfactory (0 points)	POINTS
Algorithm	To understand basic algorithms				___/3
	To write algorithms using linked structures such as List, trees, and graphs				___/3
	To apply and implement learned algorithm design techniques to solve problems				___/3
Data Structures	To describe the usage of various data structures				___/3
	To design and apply appropriate data structures for solving problems				___/3
	To understand how to choose the appropriate data structure to solve a problem				___/3
Participation	Often asks questions and shows interest to deepen knowledge				___/3
Posture	Always willing to help peers.				___/3
	Completes All class assignments.				
	Often helps peers after finishing a class assignment.				
TOTAL POINTS:					___/24