

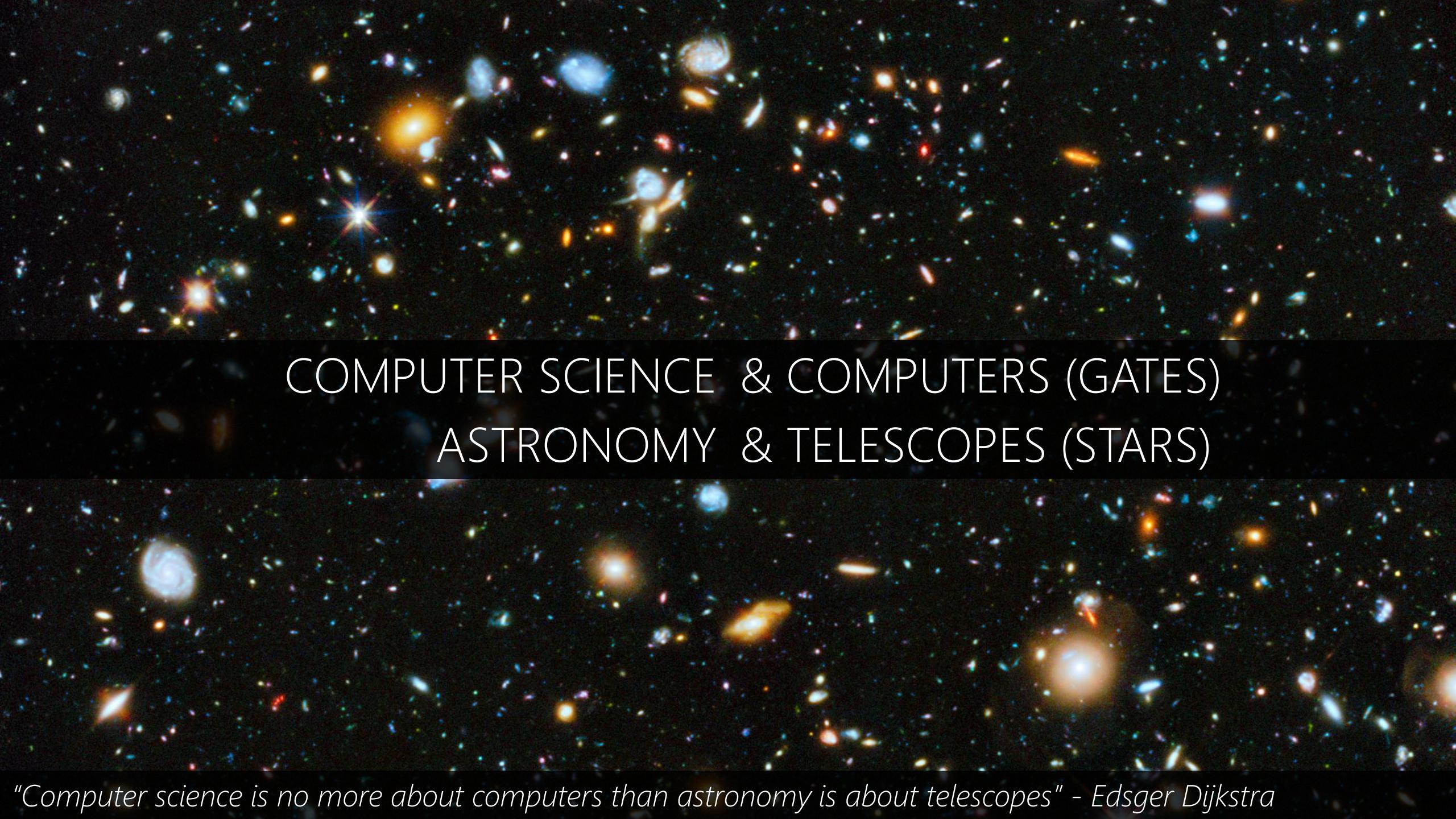
<https://hubblesite.org/contents/news-releases/2014/news-2014-27.html>



HUBBLE UNVEILS ITS MOST COLORFUL VIEW OF THE UNIVERSE
(ZOOM AND PAN)

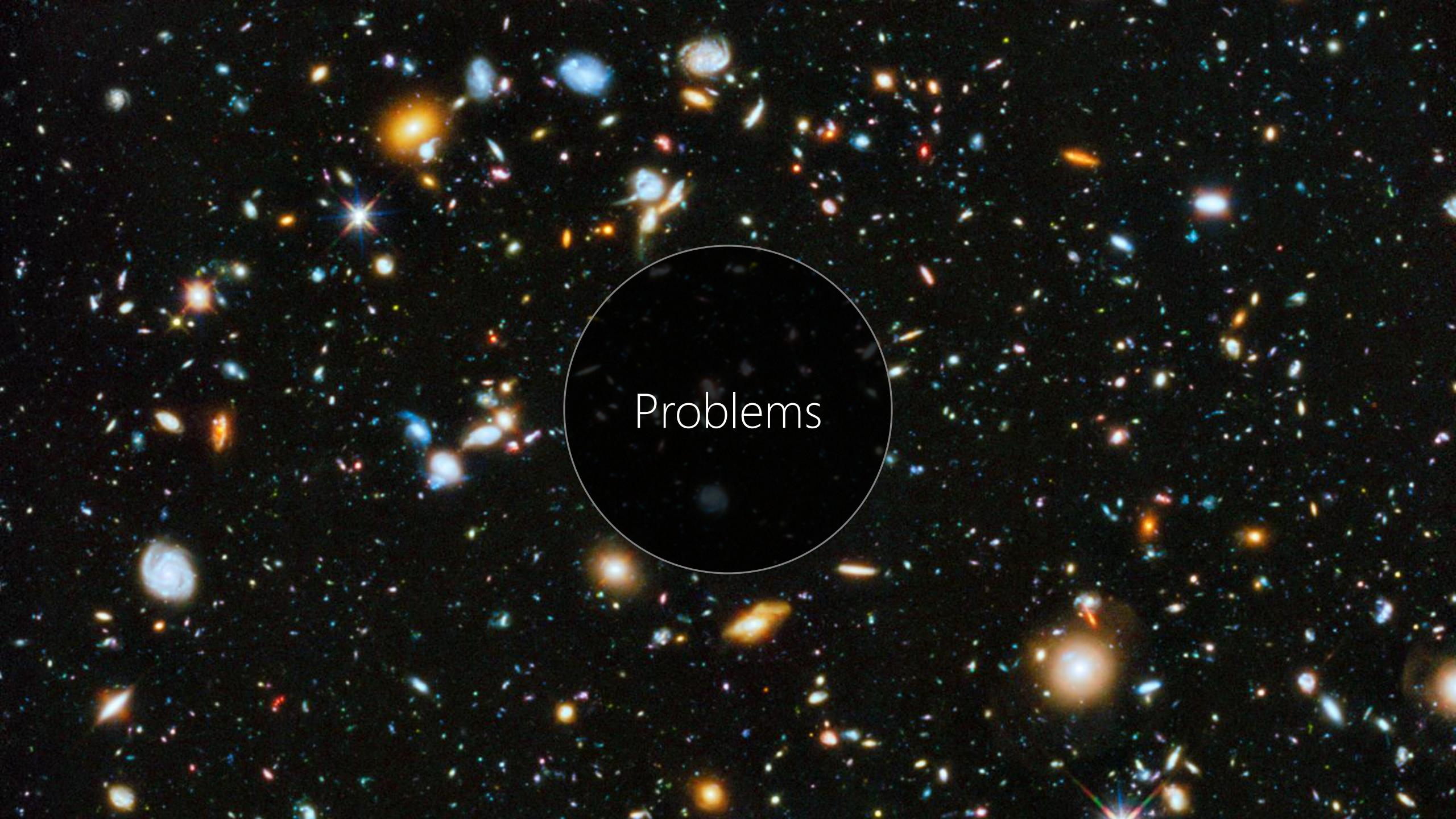
BROWSE THE LINK BELOW AND PLAY!



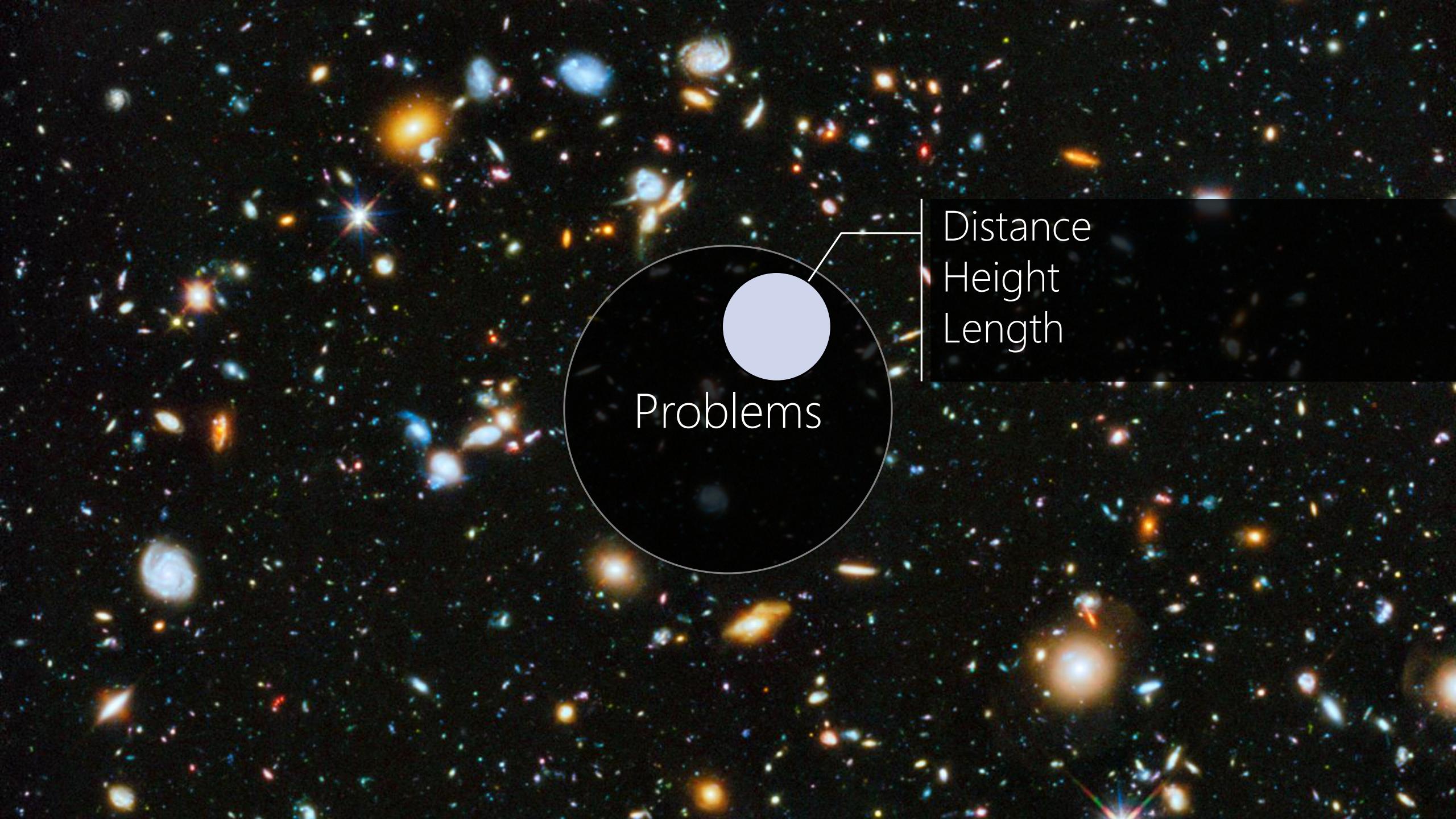


COMPUTER SCIENCE & COMPUTERS (GATES)
ASTRONOMY & TELESCOPES (STARS)

"Computer science is no more about computers than astronomy is about telescopes" - Edsger Dijkstra

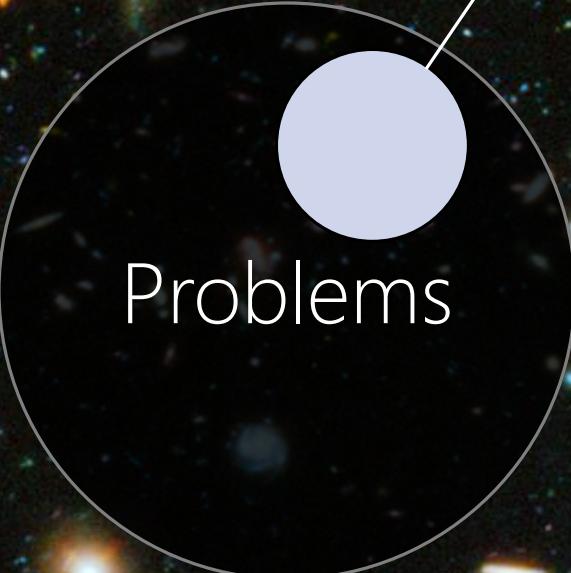


Problems

The background of the image is a deep space photograph filled with numerous galaxies of various sizes and colors, ranging from small blue and white dots to large, bright, multi-colored elliptical structures.

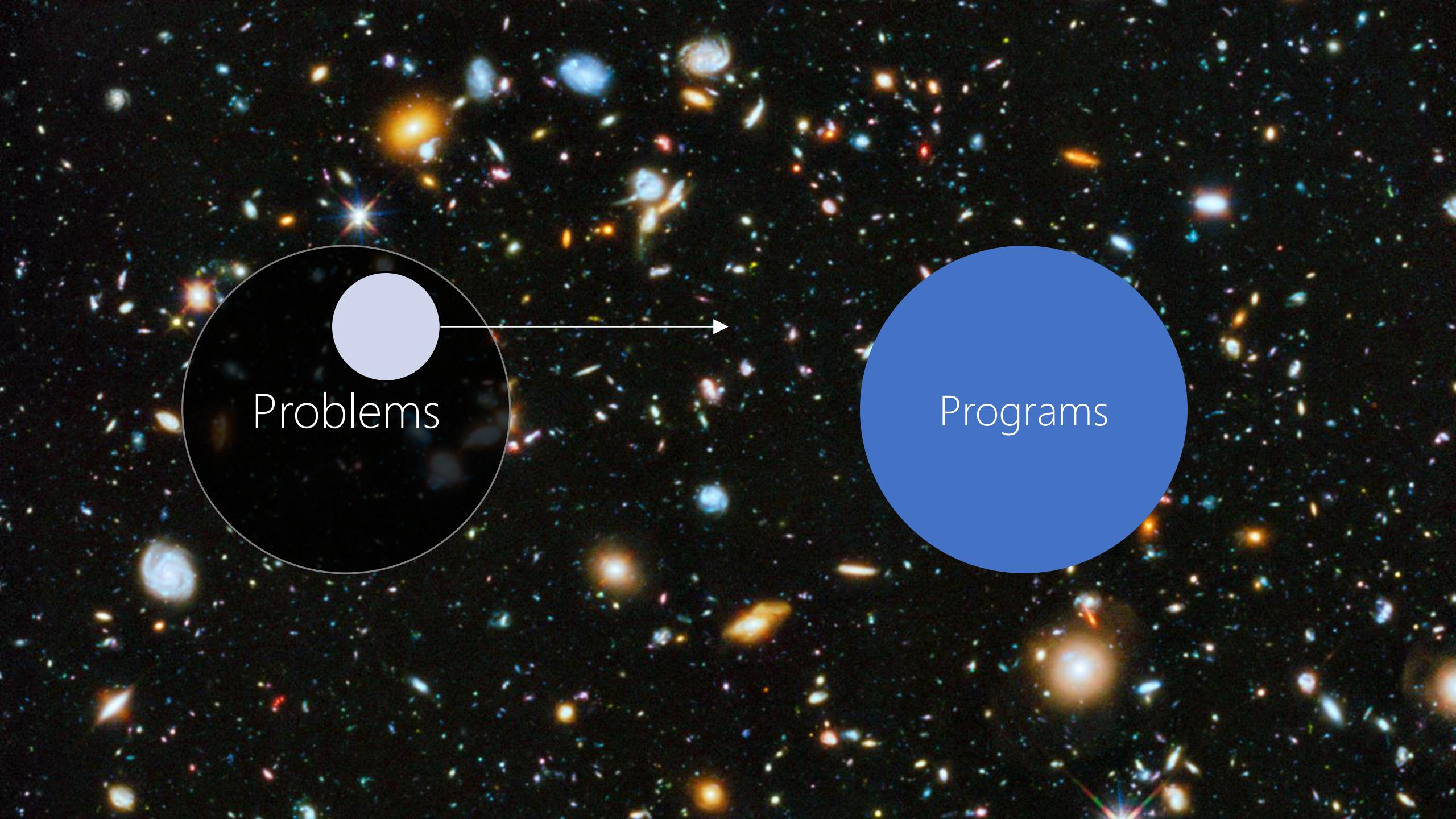
Distance
Height
Length

Problems



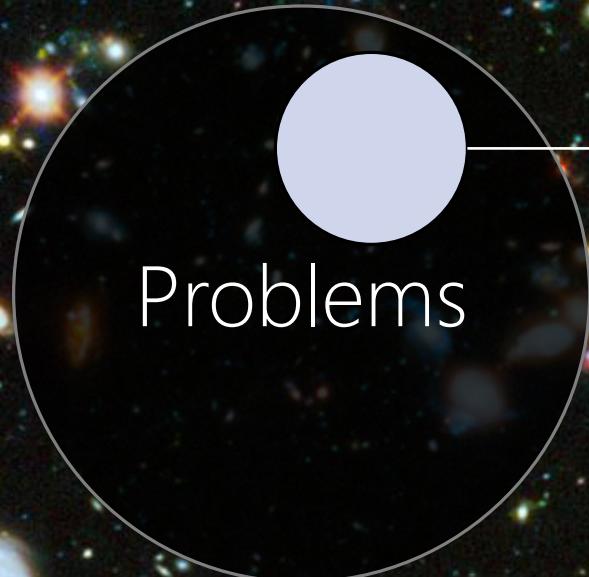
Problems

Computable
Theory of Automata
Theory of Computation
Theory of Computer Science

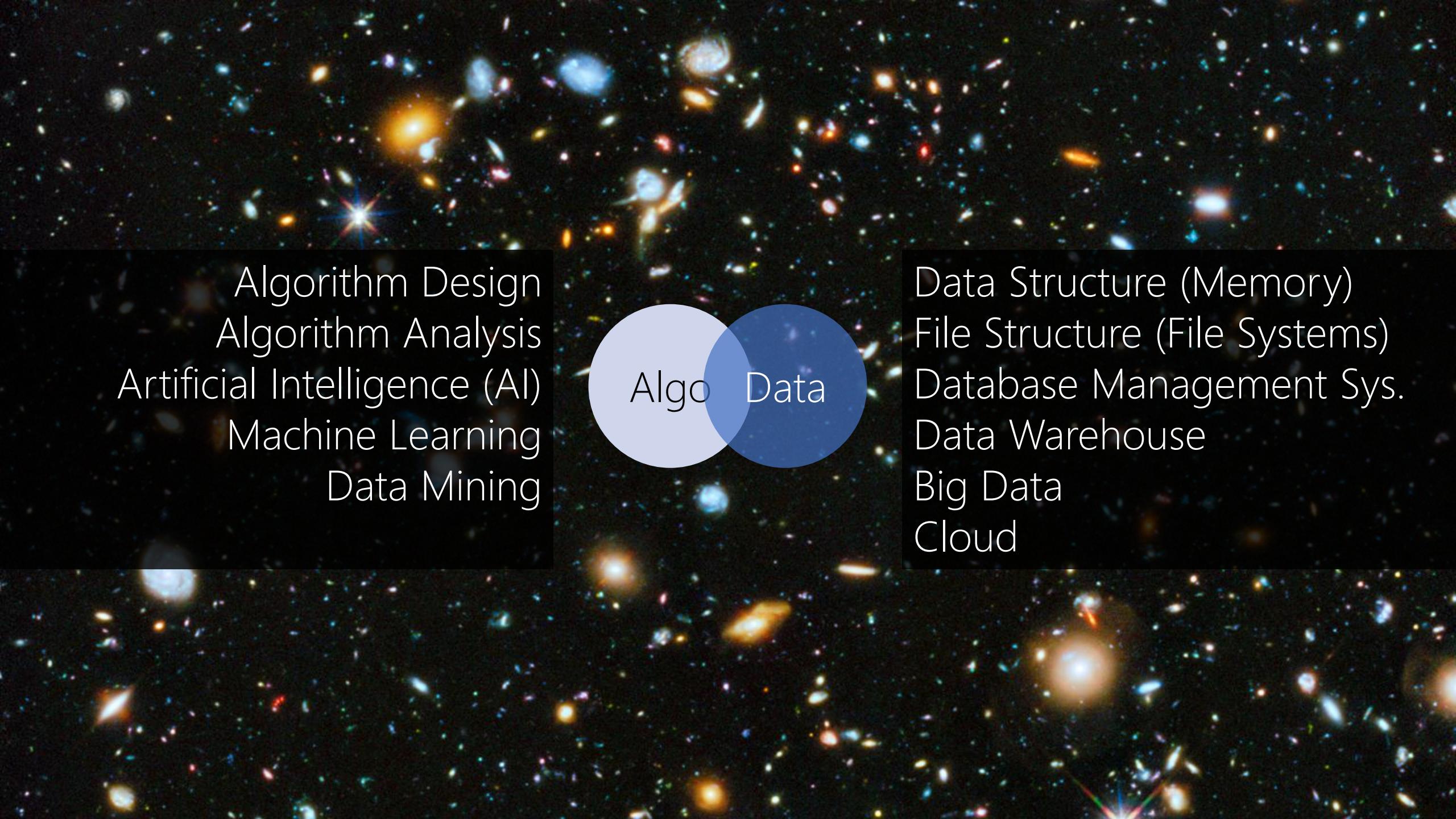


Problems

Programs



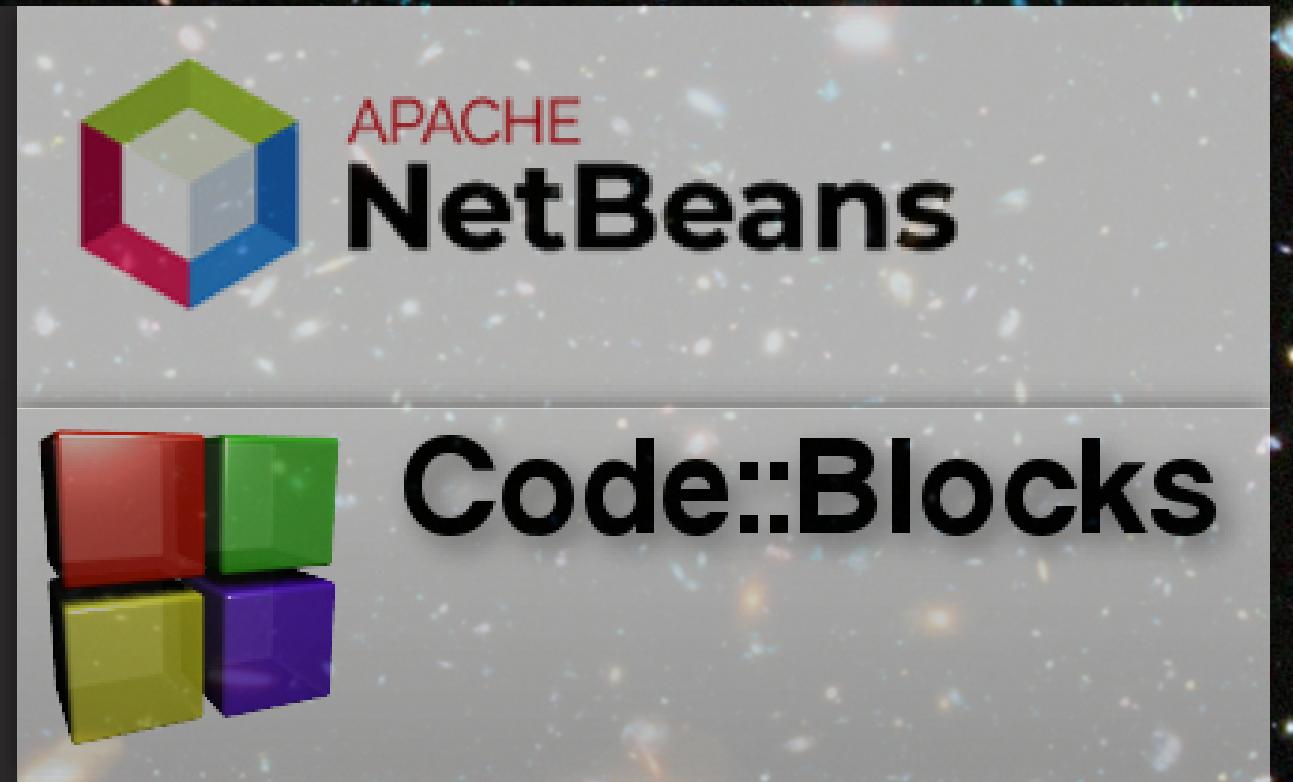
Algorithm Data

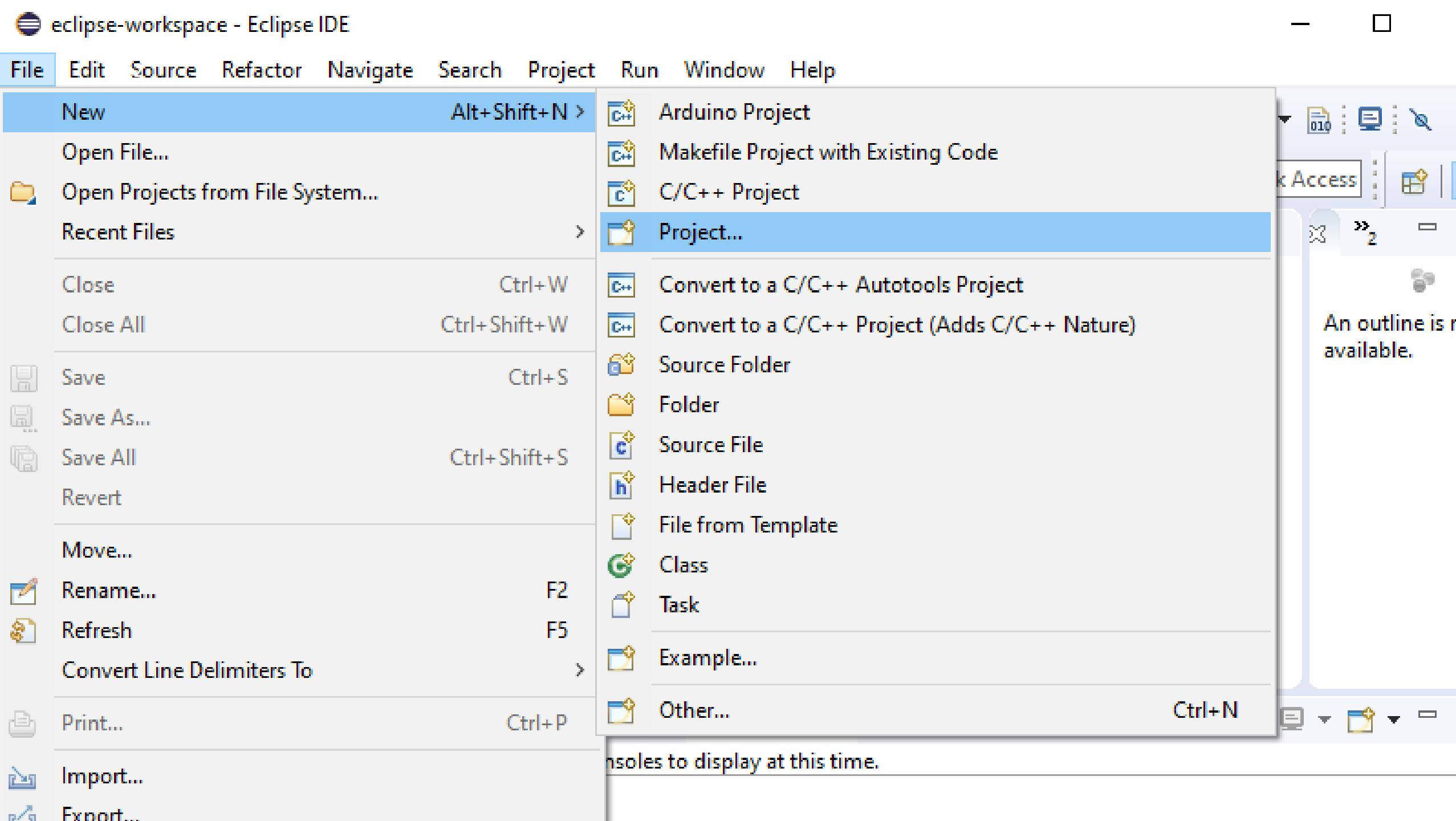


Algorithm Design
Algorithm Analysis
Artificial Intelligence (AI)
Machine Learning
Data Mining



Data Structure (Memory)
File Structure (File Systems)
Database Management Sys.
Data Warehouse
Big Data
Cloud





Select a wizard

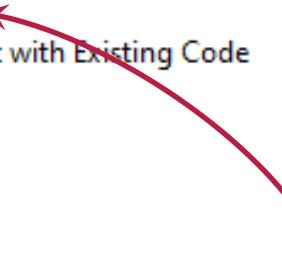
Create a new C++ project



Wizards:

type filter text

- >  General
- >  C/C++
 -  Arduino Project
 -  C Project
 -  C/C++ Project
 -  C++ Project
 -  Makefile Project with Existing Code
- >  RPM
- >  Tracing
- >  Examples



< Back

Next >

Finish

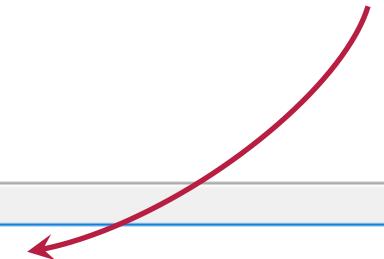
Cancel

C++ Project

Create C++ project of selected type



Project name: COMP2650_W01_Simple_Addition_hfani

 Use default location

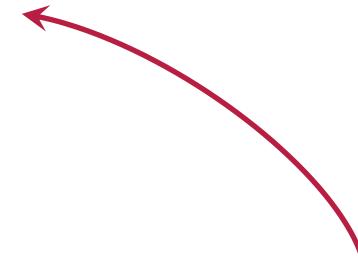
Location: C:\Users\hfani\Documents\eclipse-workspace\COMP2650_W01_Simple_Addition_hfani

Browse...

Choose file system: default

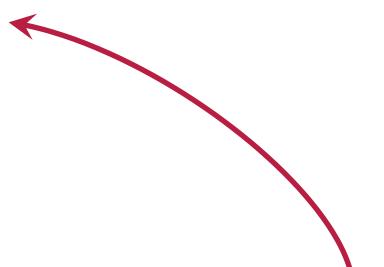
Project type:

- > GNU Autotools
- < Executable
 - Empty Project
 - Hello World C++ Project
- > Shared Library
- > Static Library
- > Makefile project



Toolchains:

- Cross GCC
- Cygwin GCC
- MinGW GCC

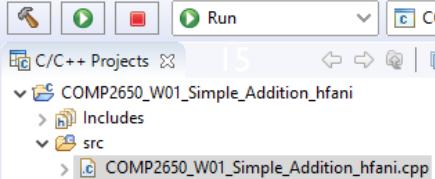
 Show project types and toolchains only if they are supported on the platform

< Back

Next >

Finish

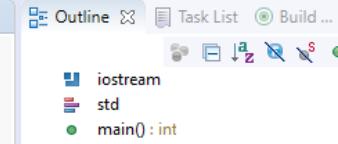
Cancel



```
1 //=====
2 // Name      : COMP2650_W01_Simple_Addition_hfani.cpp
3 // Author    : hfani@uwindsor.ca
4 // Version   :
5 // Copyright :
6 // Description : Hello World in C++, Ansi-style
7 //=====

8

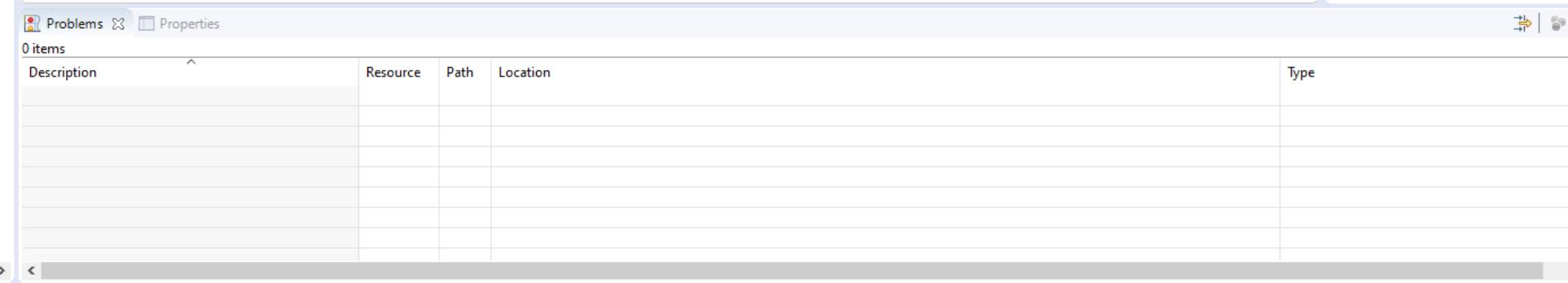
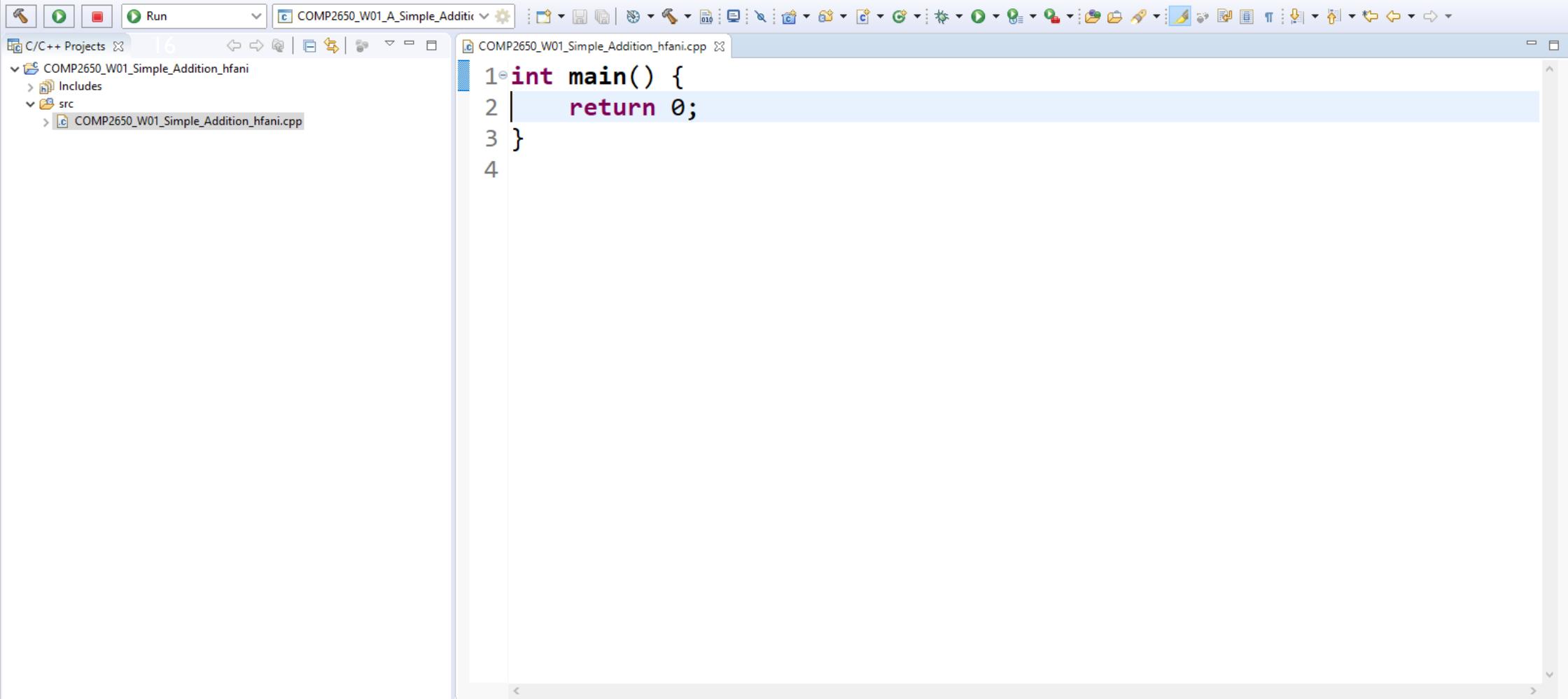
9 #include <iostream>
10 using namespace std;
11
12 int main() {
13     cout << "" << endl; // prints
14     return 0;
15 }
16 |
```



Problems X Properties

0 items

Description	Resource	Path	Location	Type
-------------	----------	------	----------	------



The screenshot shows the Eclipse IDE interface with the following details:

- Toolbar:** Standard Eclipse toolbar with icons for file operations, run, search, and preferences.
- Project Explorer:** Shows a C/C++ project named "COMP2650_W01_Simple_Addition_hfani". The "src" folder contains the file "COMP2650_W01_Simple_Addition_hfani.cpp". Other folders like "Binaries" and "Includes" are also listed.
- Code Editor:** The main window displays the source code for "COMP2650_W01_Simple_Addition_hfani.cpp". The code defines a function "Add" that adds two integers and a "main" function that prompts the user for two numbers, adds them, and prints the result.
- Outline View:** Located on the right side, it shows the symbols defined in the current file: "stdio.h", "Add(int, int) : int", and "main() : int".

```
1 #include <stdio.h>
2 int Add(int a, int b){
3     int result = a + b;
4     return result;
5 }
6 int main() {
7
8     int number1;
9     int number2;
10    printf("Enter two numbers:\n");
11
12    scanf("%d", &number1);
13    scanf("%d", &number2);
14
15    int sum = Add(number1, number2);
16
17    printf("%d + %d = %d", number1, number2, sum);
18
19    return 0;
20 }
```

Problems Console Properties

CDT Build Console [COMP2650_W01_Simple_Addition_hfani]

Finished building target: COMP2650_W01_Simple_Addition_hfani.exe

11:50:59 Build Finished. 0 errors, 0 warnings. (took 1s.147ms)

```
2 int Add(int a, int b){  
3     int result = a + b;  
4     return result;  
5 }
```

```
6  
7     int number1;  
8     int number2;  
9     printf("Enter first Number : ");  
10    scanf("%d", &number1);  
11    printf("Enter second Number : ");  
12    scanf("%d", &number2);  
13  
14    int sum = Add(number1, number2);  
15  
16    printf("Sum = %d + %d = %d", number1, number2, sum);  
17  
18    return 0;  
19
```

```
Process finished with exit code 0
```

```
03:58:09 Build Finished, 0 errors, 0 warnings. (took 0s.047ms)
```

```
14
15     int sum = Add(number1, number2);
16
```

```
17     System.out.println("Sum of " + number1 + " and " + number2 + " is " + sum);
```

```
18
19 }
```

```
Process finished with exit code 0
```

```
13:58:09 Build Finished, 0 errors, 0 warnings. (took 0s.047ms)
```

eclipse-workspace - COMP2650_W01_Simple_Addition_hfani/src/COMP2650_W01_Simple_Addition_hfani.cpp - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Run C/C++ Projects COMP2650_W01_Simple_Addition_hfani.cpp

src

```
1 #include <stdio.h>
2 int Add(int a, int b){
3     int result = a + b;
4     return result;
5 }
6 int main() {
7
8     int number1;
9     int number2;
10    printf("Enter two numbers:\n");
11
12    scanf("%d", &number1);
13    scanf("%d", &number2);
14
15    int sum = Add(number1, number2);
16
17    printf("%d + %d = %d", number1, number2, sum);
18
19    return 0;
20 }
```

Outline Task List Build ...

stdio.h

Add(int, int) : int

main() : int

Problems Console Properties

CDT Build Console [COMP2650_W01_Simple_Addition_hfani]

Finished building target: COMP2650_W01_Simple_Addition_hfani.exe

11:50:59 Build Finished. 0 errors, 0 warnings. (took 1s.147ms)

Writable Smart Insert 20 : 2



C/C++ Projects COMP2650_W01_Simple_Addition_hfani
COMP2650_W01_Simple_Addition_hfani Binaries
COMP2650_W01_Simple_Addition_hfani.exe - [amd64/le]
Includes
src COMP2650_W01_Simple_Addition_hfani.cpp
Debug

```
1 #include <stdio.h>
2 int Add(int a, int b){
3     int result = a + b;
4     return result;
5 }
6 int main() {
7
8     int number1;
9     int number2;
10    printf("Enter two numbers:\n");
11
12    scanf("%d", &number1);
13    scanf("%d", &number2);
14
15    int sum = Add(number1, number2);
16
17    printf("%d + %d = %d", number1, number2, sum);
18
19    return 0;
20 }
```

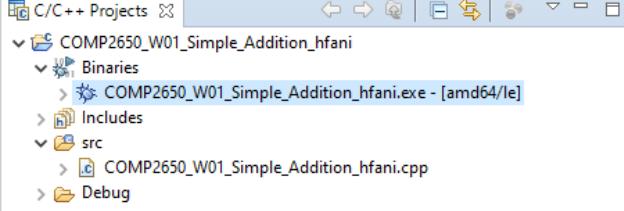
Problems Console Properties

CDT Build Console [COMP2650_W01_Simple_Addition_hfani]

Finished building target: COMP2650_W01_Simple_Addition_hfani.exe

11:50:59 Build Finished. 0 errors, 0 warnings. (took 1s.147ms)

Outline Task List Build ...
stdio.h
Add(int, int) : int
main() : int



COMP2650_W01_Simple_Addition_hfani.cpp

```
1 #include <stdio.h>
2 int Add(int a, int b){
3     int result = a + b;
4     return result;
5 }
6 int main()
7 {
8     int n1, n2;
9     printf("Enter first number: ");
10    scanf("%d", &n1);
11    printf("Enter second number: ");
12    scanf("%d", &n2);
13    int sum = Add(n1, n2);
14    printf("Sum of %d and %d is %d\n", n1, n2, sum);
15    return 0;
16 }
```

Run As

Select a way to run 'COMP2650_W01_A_Simple_Addition_hfani':

- C/C++ Container Application
- Local C/C++ Application**

Description

Runs a local C/C++ application

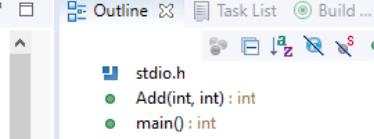
OK Cancel

CDT Build Console [COMP2650_W01_Simple_Addition_hfani]

FINISHED BUILD

11:50:59 Build Finished. 0 errors, 0 warnings. (took 1s.147ms)

addition_ntan1.exe



Enter two integers:

12

35

12 + 35 = 47



Enter two integers:

999999

1

999999 + 1 = 1000000



Enter two integers:

99999999999999

1



Enter two integers:

99999999999999

1

-1530494977 + 1 = -1530494976



Enter two integers:

99999999999999

1

-1530494977 + 1 = -1530494976



The screenshot shows the Eclipse IDE interface for a C/C++ project named "COMP2650_W01_Simple_Addition_hfani". The project structure is visible in the left sidebar, showing a "src" folder containing "COMP2650_W01_Simple_Addition_hfani.cpp". The main window displays the source code for this file:

```
1 #include <stdio.h>
2 #include <limits.h>
3 int Add(int a, int b){
4     int result = a + b;
5     return result;
6 }
7 int main() {
8
9     int number1;
10    int number2;
11    printf("Enter two numbers between %d and %d:\n", INT_MIN, INT_MAX);
12
13    scanf("%d", &number1);
14    scanf("%d", &number2);
15
16    int sum = Add(number1, number2);
17
18    printf("%d + %d = %d", number1, number2, sum);
19
20    return 0;
21 }
22
```

The code implements a simple addition program. It includes headers for stdio.h and limits.h, defines an Add function, and a main function that prompts the user for two integers, adds them using the Add function, and prints the result.

CDT Build Console [COMP2650_W01_Simple_Addition_hfani]
Finished building target: COMP2650_W01_Simple_Addition_hfani.exe

12:21:03 Build Finished. 0 errors, 0 warnings. (took 1s.164ms)

Enter two integers between -2147483648 and 2147483647:

12

15

$$12 + 15 = 27$$



Enter two integers between -2147483648 and 2147483647:

2147483647

1



Enter two integers between -2147483648 and 2147483647:

2147483647

1

2147483647 + 1 = -2147483648



Enter two integers between -2147483648 and 2147483647:

2147483647

2

2147483647 + 2 = -2147483647



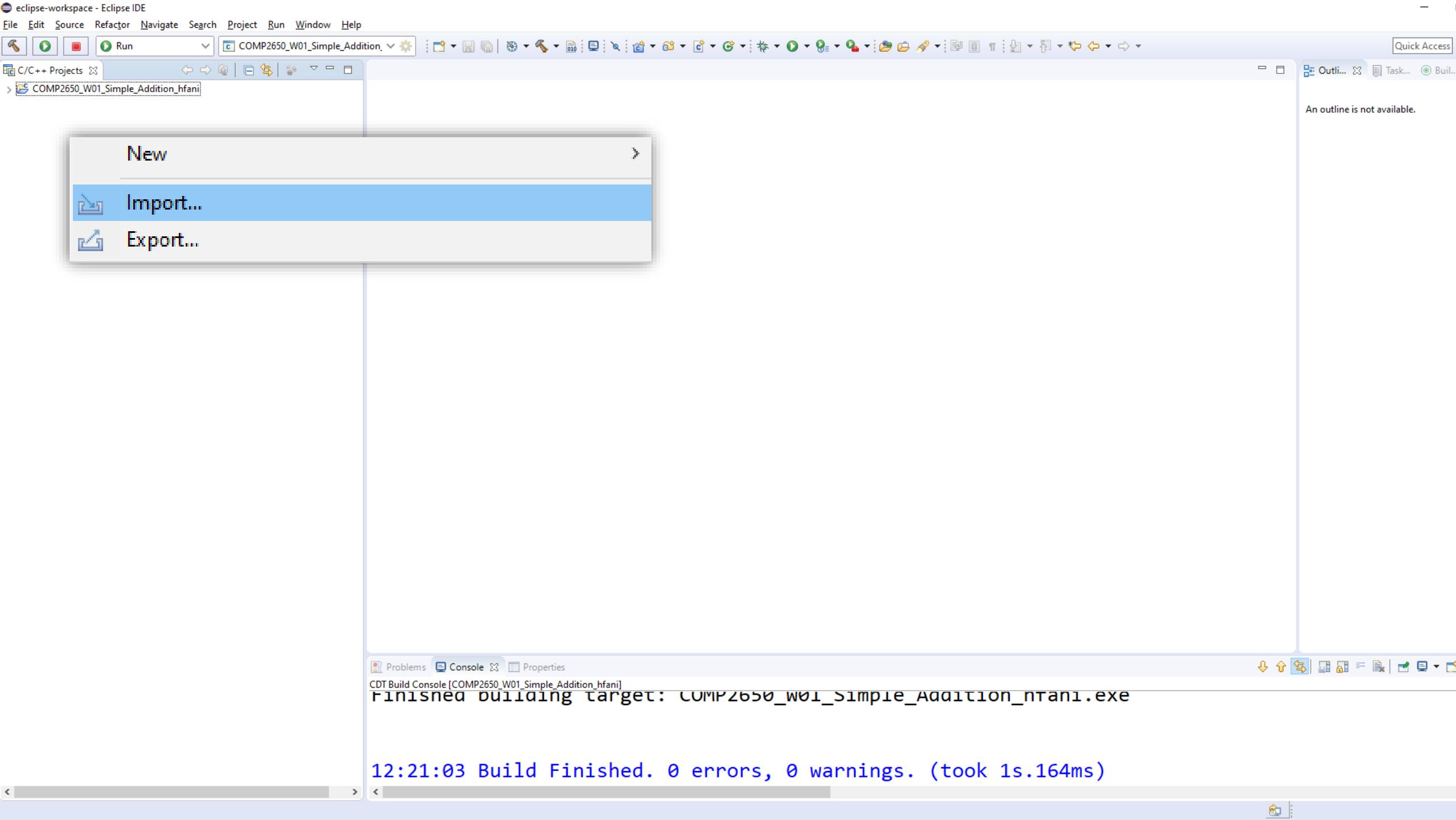
Enter two integers:

2147483647

1

2147483647 + 1 = 2147483648







Select

Create new projects from an archive file or directory.



Select an import wizard:

|

- ▼ General
 - Archive File
 - Existing Projects into Workspace
 - File System
 - Preferences
 - Projects from Folder or Archive
- > C/C++
- > Git
- > Install
- > Oomph
- > RPM
- > Run/Debug



< Back

Next >

Finish

Cancel



Import Projects

Select a directory to search for existing Eclipse projects.

 Select root directory: C:\Users\hfani\Documents\eclipse-workspace\COMP2650_W01_Large_Addition_hfani Select archive file:

Projects:

 COMP2650_W01_Large_Addition_hfani (C:\Users\hfani\Documents\eclipse-workspace\COMP2650_W01_Large_Addition_hfani)

Options

 Search for nested projects Copy projects into workspace Close newly imported projects upon completion Hide projects that already exist in the workspace

Working sets

 Add project to working sets Working sets: 

< Back

Next >

Finish

Cancel

1 //Chitra Nayal, Mithun Kumar, Shivam Kushik, 04-06-2019, Sum of two large numbers
2 //Retrieved from <https://www.geeksforgeeks.org/sum-two-large-numbers/>

```
using namespace std;  
  
string add(string s1, string s2) {  
    int n1 = s1.size(), n2 = s2.size();  
    if (n1 < n2) return add(s2, s1);  
    if (n1 == 0) return "0";  
    if (n1 == 1) return s1 + s2;  
    int carry = 0, sum;  
    string ans = "";  
    for (int i = n1 - 1; i >= 0; i--) {  
        sum = (s1[i] - '0') + (s2[n1 - 1 - i] - '0') + carry;  
        if (sum >= 10) {  
            carry = 1;  
            sum -= 10;  
        } else  
            carry = 0;  
        ans += to_string(sum);  
    }  
    if (carry) ans += to_string(carry);  
    reverse(ans.begin(), ans.end());  
    return ans;  
}
```

```
int carry = 0;
for (int i = 0; i < n1; i++) {
    // Do school mathematics, compute sum of current digits and carry
    int sum = ((a[i] - '0') + (b[i] - '0') + carry);
    result.push_back(sum % 10 + '0');
    // Calculate carry for next step
    carry = sum / 10;
}
```

Enter two integers:

2147483647

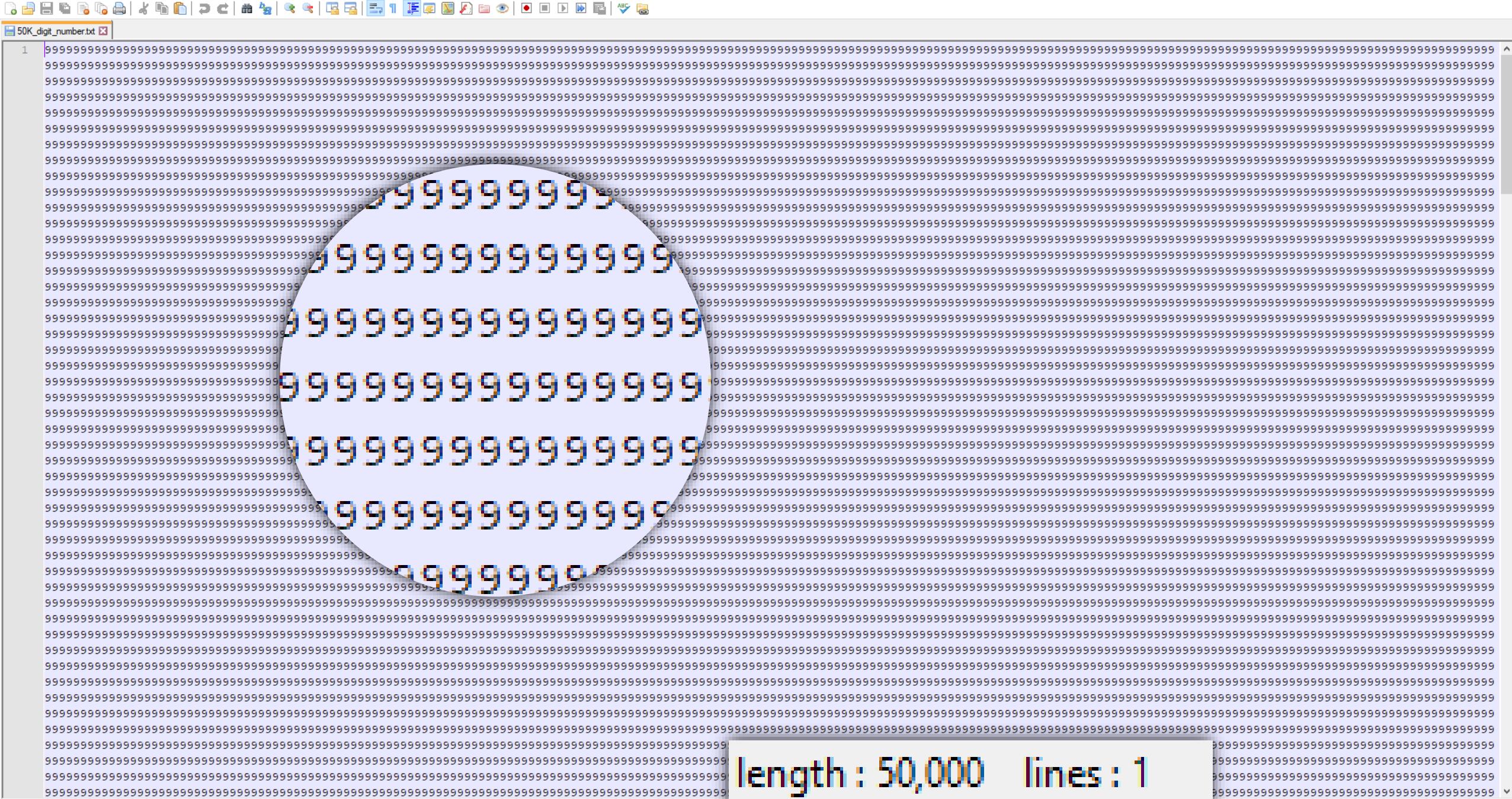
1

2147483647 + 1 = 2147483648



Enter two integers:

1



BUILD SUCCESSFUL (total time: 715ms)

Enter two integers:

999...99999 => 50,000 digits

999...99999 => 50,000 digits

999...99999 + 999...99999 = ?



BUILD SUCCESSFUL (total time: 715ms)

Enter two integers:

999...99999 => 50,000 digits

999...99999 => 50,000 digits

999...99999 + 999...99999 = 1999 ... 9998 => 50,001 digits



BUILD SUCCESSFUL (total time: 715ms)

Enter two integers:

999...99999 => 50,000 digits

999...99999 => 50,000 digits

999...99999 + 999...99999 = 1999 ... 9998 => 50,001 digits

Total time = ?



BUILD SUCCESSFUL (total time: 715ms)

Enter two integers:

999...99999 => 50,000 digits

999...99999 => 50,000 digits

999...99999 + 999...99999 = 1999 ... 9998 => 50,001 digits

Total time = 0.000000



BUILD SUCCESSFUL (total time: 715ms)

Enter two integers:

999...99999 => 50,000 digits

999...99999 => 50,000 digits

How many iterations:

?

Total time = ?



Matrix

BUILD SUCCESSFUL (total time: 715ms)

Enter two integers:

999...99999 => 50,000 digits

999...99999 => 50,000 digits

How many iterations:

1

Total time = 0.000000



BUILD SUCCESSFUL (total time: 715ms)

Enter two integers:

999...99999 => 50,000 digits

999...99999 => 50,000 digits

How many iterations:

100

Total time = ?



BUILD SUCCESSFUL (total time: 715ms)

Enter two integers:

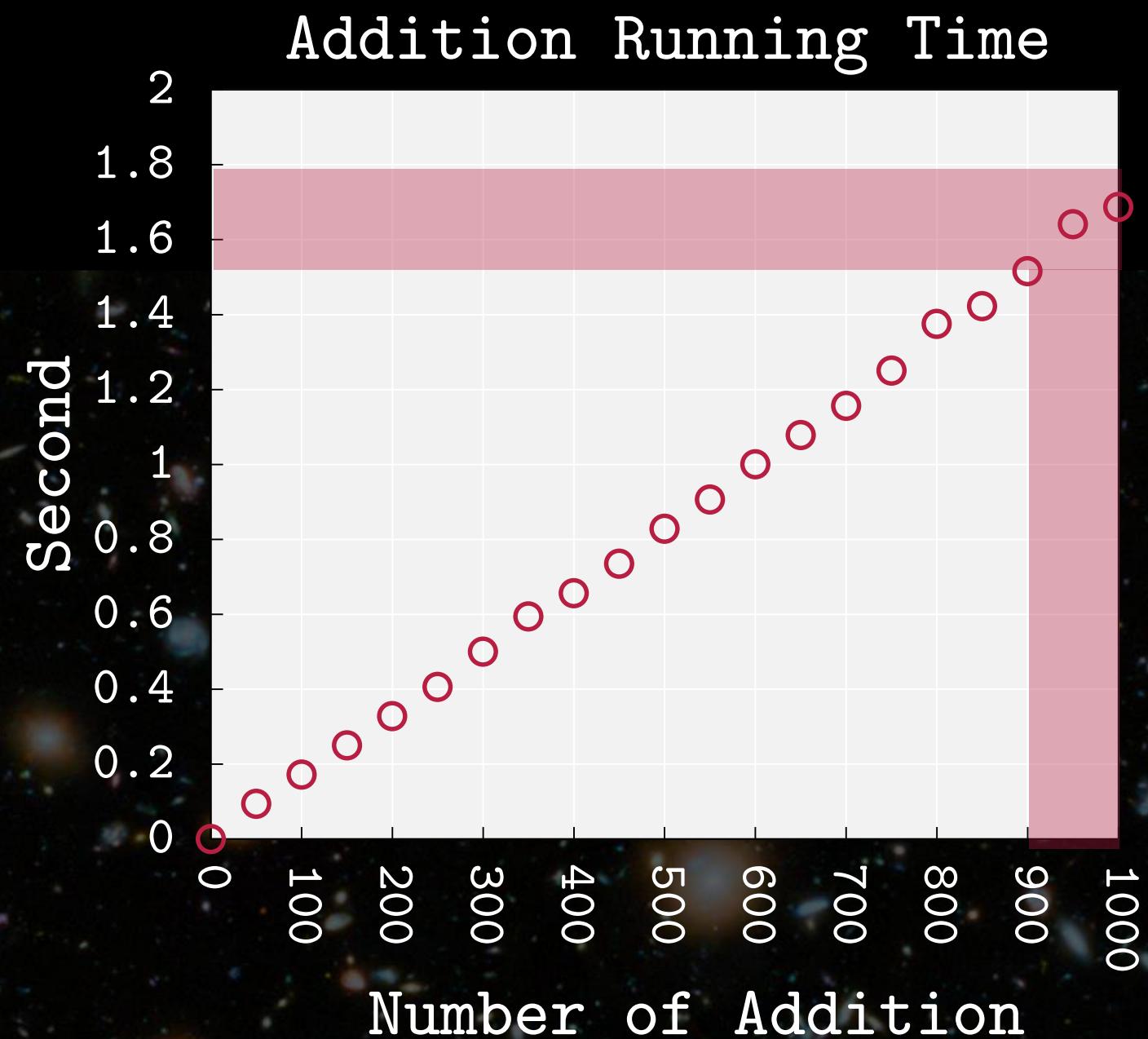
999...99999 => 50,000 digits

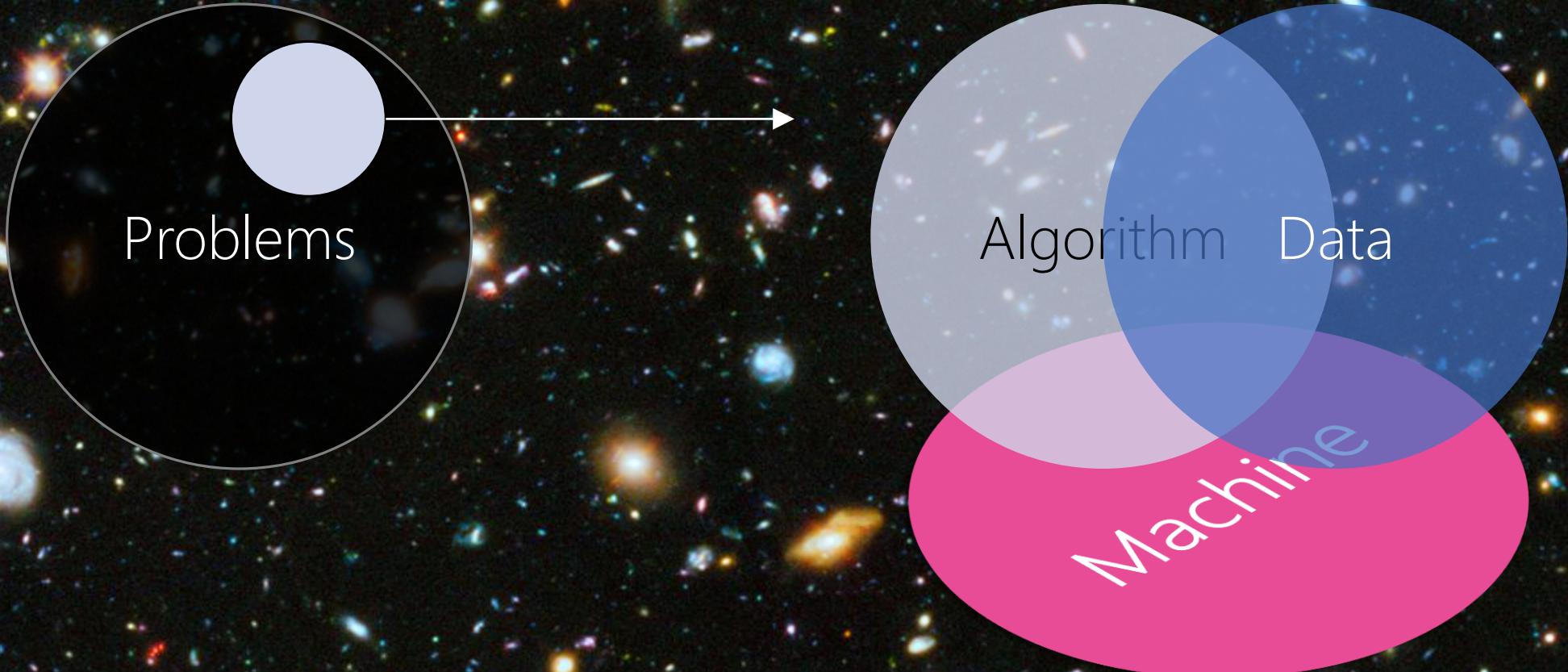
999...99999 => 50,000 digits

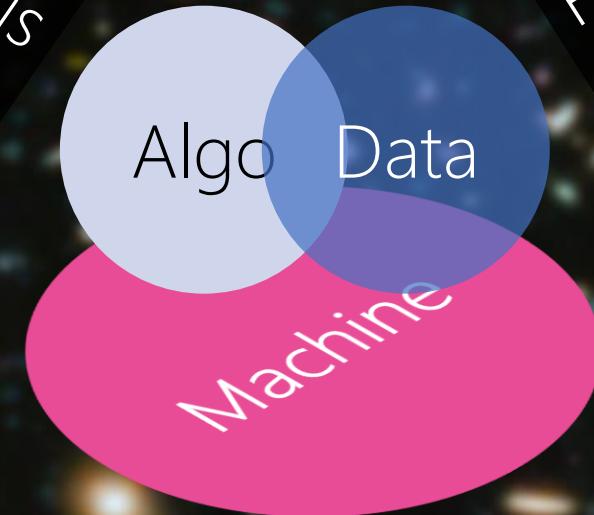
How many iterations:

1,2,3,...,900,..., 1000

Total time =







Algorithm Design
Algorithm Analysis
Artificial Intelligence (AI)
Machine Learning
Data Mining

Data Structure
File Structure
Database Management Sys.
Data Warehouse
Big Data
Cloud

Digital Design (Logic Circuits)
Computer Architecture
Microprocessor Programming



Algorithm Design
Algorithm Analysis
Artificial Intelligence (AI)
Machine Learning
Data Mining

Data Structure
File Structure
Database Management Sys.
Data Warehouse
Big Data
Cloud

Digital Design (Logic Circuits)
Computer Architecture
Microprocessor Programming

DIGITAL DESIGN

aka. Logic Circuits

Fall 2020

DIGITAL DESIGN

aka. Logic Circuits

blackboard.uwindsor.ca → [COMP2650-1-R-2020F-a Computer Architecture I:Digital Design](#)

BLACKBOARD TOUR

Lectures → How To Find → [COMP2650-1-R-2020F-a Blackboard Tutorial Video 1: Course Homepage Tour](#)

LEARNING OUTCOME

aka. Learning Objectives

Learning Outcome → [COMP2650 Computer Architecture I Digital Design Learning Outcome.pdf](#)



UNIVERSITY OF WINDSOR

PROGRAM DEVELOPMENT COMMITTEE
COURSE LEARNING OUTCOMES FORM

COURSE NUMBER AND TITLE: 0360-265. Computer Architecture I: Digital Design

Please complete the following table. State the specific learning outcomes that make up the goal of the course (what will students know and be able to do at the end of this course?) and link the learning outcomes to the Characteristics of a University of Windsor Graduate outlined in "To Greater Heights" by listing them in the appropriate rows.

Please note that a learning outcome may link to more than one of the specified Characteristics of a University of Windsor Graduate, and that a single course might not touch on each of the Characteristics. Each University of Windsor program should produce graduates that are able to demonstrate each of the nine characteristics approved in To Greater Heights.

Information on learning outcomes is appended to this form (Appendix A). Proposers are also strongly encouraged to contact the Office of the Vice-Provost, Teaching and Learning or the Centre for Teaching and Learning, for assistance with the articulation of learning outcomes.

Course Learning Outcomes (see Appendix A for more on learning outcomes)	Characteristics of a University of Windsor Graduate
At the end of the course, the successful student will know and be able to:	A U of Windsor graduate will have the ability to demonstrate:
<ul style="list-style-type: none">• Analyze and design combinatorial & sequential circuits.• Explain how a computer system works.	A. the acquisition, application and integration of knowledge
<ul style="list-style-type: none">• Retrieve and evaluate information about the performance and speed of different processors.	B. research skills, including the ability to define problems and access, retrieve and evaluate information (information literacy)
<ul style="list-style-type: none">• Analyze and design combinatorial & sequential circuits.• Design arithmetic circuits, logic circuits and shifting circuits, according to specifications.• Design Arithmetic Logic Unit for processors.• Develop control circuits for a processor.	C. critical thinking and problem-solving skills
<ul style="list-style-type: none">• Explain how a computer system works.	D. literacy and numeracy skills
	E. responsible behaviour to self, others and society
	F. interpersonal and communications skills



Course Learning Outcomes

(see [Appendix A](#) for more on learning outcomes)

At the end of the course, the successful student will know and be able to:

Characteristics of a University of Windsor Graduate

A U of Windsor graduate will have the ability to demonstrate:

- Analyze and design combinatorial & sequential circuits.
- Explain how a computer system works.

A. the acquisition, application and integration of knowledge

- Retrieve and evaluate information about the performance and speed of different processors.
- B. research skills, including the ability to define problems and access, retrieve and evaluate information (information literacy)



- Analyze and design combinatorial & sequential circuits.
- Design arithmetic circuits, logic circuits and shifting circuits, according to specifications.
- Design Arithmetic Logic Unit for processors.
- Develop control circuits for a processor.

C. critical thinking and problem-solving skills

- Explain how a computer system works.

D. literacy and numeracy skills

	E. responsible behaviour to self, others and society
	F. interpersonal and communications skills

COURSE OUTLINE

aka. Course Syllabus

Outline (Syllabus) → [COMP2650 Computer Architecture I Digital Design Course Outline.pdf](#)



**School of Computer Science
Faculty of Science**

**COMP-2650: Computer Architecture I: Digital Design
Fall 2020**

Course Outline (Syllabus) v1.0

Digital systems are indispensable and the foremost means of technological progress. We refer to the present period as the digital age. Digital systems control traffic, power generation, industrial control, spacecraft guidance, medical treatment, and many other aspects of our lives. A digital device surrounds us, including digital telephones (Smartphones), digital cameras, and, of course, digital computers. They work easily, they involve precise execution, and they can do math at the hardware level indeed.

s) v1.0

st means

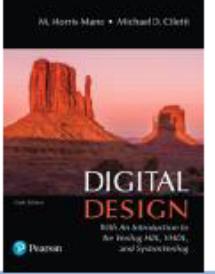
Course Type	Required
Program Level	COMP 2
Prerequisite	COMP-1400: Introduction to Algorithms and Programming
Required for	COMP-2660: Computer Architecture II: Microprocessor Programming COMP-3150: Database Management Systems COMP-3300: Operating Systems Fundamentals COMP-3500: Introduction to Multimedia Systems COMP-3670: Computer Networks

Course Type	Required
Program Level	COMP 2
Prerequisite	COMP-1400: Introduction to Algorithms and Programming
Required for	COMP-2660: Computer Architecture II: Microprocessor Programming COMP-3150: Database Management Systems COMP-3300: Operating Systems Fundamentals COMP-3500: Introduction to Multimedia Systems COMP-3670: Computer Networks
Instructional Hour	3:00 lecture + 01:20 laboratory, each week.
Lecture	Mondays & Wednesdays, 08:30AM - 09:50AM, Blackboard Collaborate Ultra
Office Hour	Mondays & Wednesdays, 10:00AM - 11:00AM, Blackboard Collaborate Ultra
Homepage	blackboard.uwindsor.ca → COMP2650-1-R-2020F-a: Computer Architecture I: Digital Design

A screenshot of a Blackboard course page. The main content area displays a book resource titled "Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog, 6/E" by M. Morris R. Mano and Michael D. Ciletti. The ISBN information is also provided: ISBN-10: 0134549899, ISBN-13: 9780134549897, and the copyright information: ©2018 Pearson. To the right of the text is the front cover of the book, which features a landscape photograph of Monument Valley at sunset with the title "DIGITAL DESIGN" and subtitle "With an Introduction to the Verilog HDL, VHDL, and SystemVerilog". The background of the page shows a blurred view of the course navigation menu on the left and a list of course resources on the right.

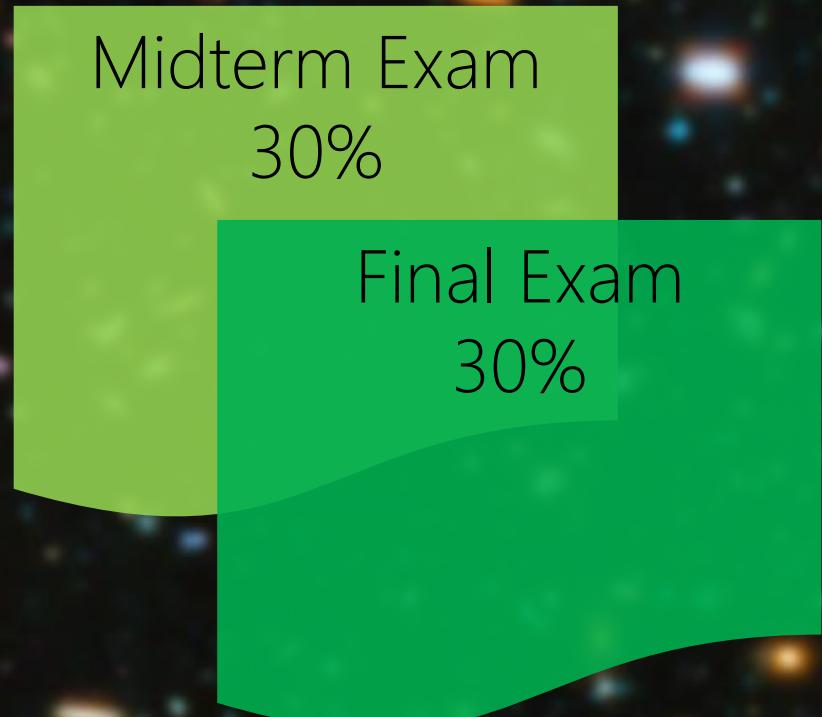
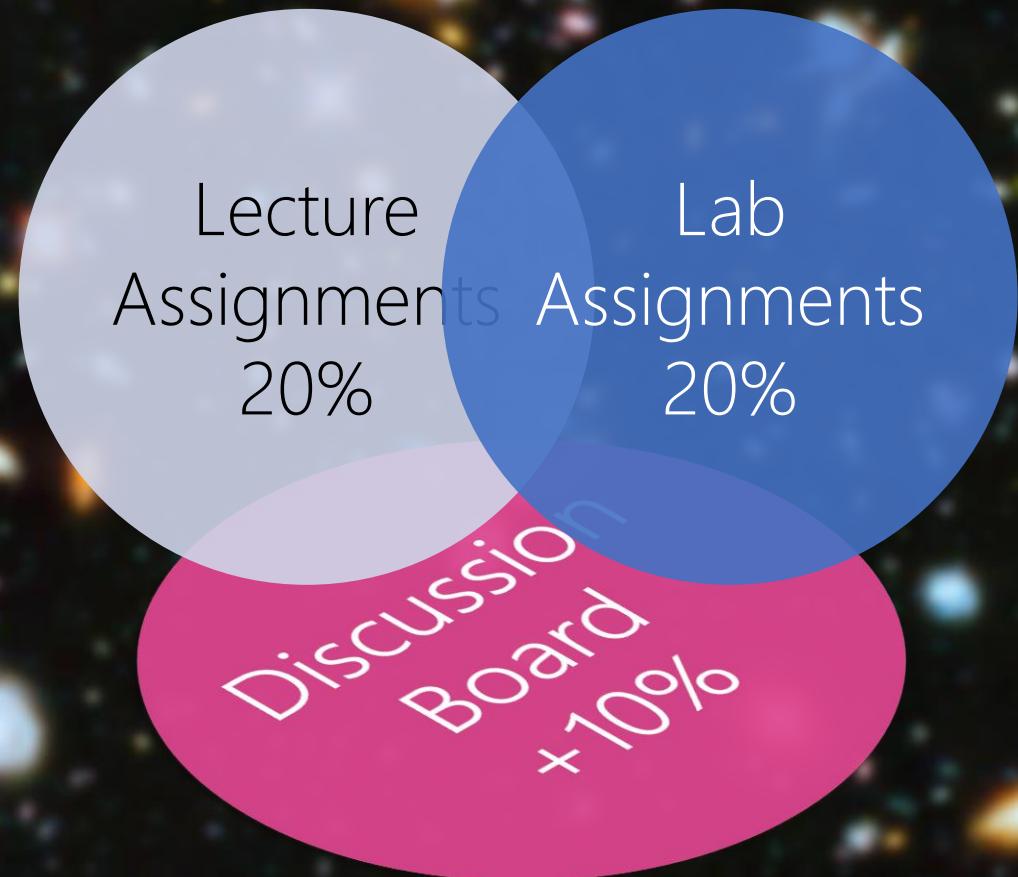
Book

Digital Design: With an Introduction to the Verilog HDL, VHDL, and SystemVerilog, 6/E
M. Morris R. Mano, Michael D. Ciletti
ISBN-10: 0134549899, ISBN-13: 9780134549897
©2018 Pearson





Marking Scheme	Lecture (Weekly) Assignments	20%
	Lab Assignments	20%
	Midterm Exam	30%
	Final Exam	30%
	Discussion Board (Bonus+)	10%
Remarks	The written reports will be assessed not only on their academic merit but also on the communication skills of the student as exhibited through the reports. To achieve a passing grade, the students must achieve at least 50% of the entire marking scheme . The students earn final course grades as per the Senate policy for Grading and Calculation of Averages and Grading Key .	



Theoretical ³	<p>W01A: Meet and Greet Course Outline. Digital Systems.</p> <p>W01B: Number Systems I Binary, Octal, and Hexa Numbers. Number-base Conversions.</p> <p>W02: Number Systems II Complements of Numbers. Signed Numbers. Binary Codes.</p> <p>W03: Boolean Algebra and Logic Gates Axiomatic Definition, Basic Theorems, and Properties of Boolean Algebra. Boolean Functions. Canonical and Standard Forms. Digital Logic Gates.</p> <p>W04: Gate-Level Minimization The Map Method. Four-Variable K-Map. Product-of-Sums Simplification.</p> <p>W05: Reading Week No Class</p> <p>W06A: Midterm Exam</p> <p>W06B: Gate-Level Minimization Don't-Care Conditions. NAND, NOR, XOR, Wired AND, Wired OR.</p> <p>W07: Combinational Logic I Adders. Subtractors. Multipliers.</p> <p>W08: Combinational Logic II Decoders. Encoders. Multiplexers. Demultiplexers.</p> <p>W09: Synchronous Sequential Logic I Sequential Circuits. Latches. Flip-Flops.</p> <p>W10: Synchronous Sequential Logic II Analysis of Clocked Sequential Circuits. State Reduction and Assignment.</p> <p>W11: Synchronous Sequential Logic III Registers and Counters. Shift Registers.</p> <p>W12: Synchronous Sequential Logic IV Ripple Counters. Synchronous Counters.</p> <p>W13: Memory and Programmable Logic RAM, Memory Decoding. Error Detection & Correction. ROM.</p> <p>W14: Final Exam Date and Time to be Announced</p>	<p>Sep. 14</p> <p>Sep. 16</p> <p>Sep. 21 & 23 Last Day Academic Add/Drop</p> <p>Sep. 28 & 30</p> <p>Oct. 05 & 07 Financial Drop Date</p> <p>Oct. 10 - 18</p> <p>Oct. 19</p> <p>Oct. 21</p> <p>Oct. 26 & 28</p> <p>Nov. 02 & 04</p> <p>Nov. 09 & 11 Midterm Exam Grade Release</p> <p>Nov. 16 & 18 Last Day Voluntarily Withdraw</p> <p>Nov. 23. & 25</p> <p>Nov. 30 & Dec. 02</p> <p>Dec. 07 & 09 Last Day: Fall 2020 Classes</p> <p>Dec. 11 – 22</p>
--------------------------	---	--

Attendance

Encouraged but **not mandatory** due to time zone accommodation in the COVID-19 pandemic era. Lecture recordings with captions along with presentation slides will be made available.

Blackboard Tour | Learning Outcome | Outline (Syllabus) | Laboratory | Discussion Board | Me | Weekly Schedule

Laboratory ⁴	<table border="1"><tr><td data-bbox="499 637 960 688">Section 51, Mondays</td><td data-bbox="960 637 2534 688">11:30AM - 12:50PM, Blackboard Collaborate Ultra</td></tr><tr><td data-bbox="499 695 960 745">Section 52, Wednesdays</td><td data-bbox="960 695 2534 745">11:30AM - 12:50PM, Blackboard Collaborate Ultra</td></tr><tr><td data-bbox="499 753 960 803">Section 53, Mondays</td><td data-bbox="960 753 2534 803">01:00PM - 02:20PM, Blackboard Collaborate Ultra</td></tr><tr><td data-bbox="499 810 960 861">Section 54, Wednesdays</td><td data-bbox="960 810 2534 861">01:00PM - 02:20PM, Blackboard Collaborate Ultra</td></tr><tr><td data-bbox="499 868 960 918">Section 55, Tuesdays</td><td data-bbox="960 868 2534 918">11:30AM - 12:50PM, Blackboard Collaborate Ultra</td></tr></table>	Section 51, Mondays	11:30AM - 12:50PM, Blackboard Collaborate Ultra	Section 52, Wednesdays	11:30AM - 12:50PM, Blackboard Collaborate Ultra	Section 53, Mondays	01:00PM - 02:20PM, Blackboard Collaborate Ultra	Section 54, Wednesdays	01:00PM - 02:20PM, Blackboard Collaborate Ultra	Section 55, Tuesdays	11:30AM - 12:50PM, Blackboard Collaborate Ultra
Section 51, Mondays	11:30AM - 12:50PM, Blackboard Collaborate Ultra										
Section 52, Wednesdays	11:30AM - 12:50PM, Blackboard Collaborate Ultra										
Section 53, Mondays	01:00PM - 02:20PM, Blackboard Collaborate Ultra										
Section 54, Wednesdays	01:00PM - 02:20PM, Blackboard Collaborate Ultra										
Section 55, Tuesdays	11:30AM - 12:50PM, Blackboard Collaborate Ultra										

1. **Equity, Diversity, and Inclusiveness (EDI):** This course, along with all its components such as lab sections are, without question, safe places for students of all races, genders, sexes, ages, sexual orientations, religions, disabilities, and socioeconomic statuses. Disrespectful attitude, sarcastic comments, offensive language, or language that could be translated as offensive and/or marginalize anyone are absolutely unacceptable. Immediate actions will be taken by the instructor to protect the safety and comfort of the students. An ethnically rich and diverse multi-cultural world should be celebrated in the classroom. The instructor, too, must treat every student equally and with the respect and compassion that all students deserve. Furthermore, UWindsor is committed to combatting sexual misconduct. All members are required to report any instances of sexual misconduct, including harassment and sexual violence, to the [Sexual Misconduct Response & Prevention Office](#) so that the victim may be provided appropriate resources and support options.
2. **Student Accessibility Services:** Students who have special needs due to legitimate medical reasons should notify the [Student Accessibility Services](#) and the instructor at the beginning of the course and before any assessment.

3. **Lab and Lecture (Weekly) Assignment:** Assignments are expected to be submitted on the assigned due date and time. Late submission is not accepted and receive zero unless a verifiable reason with appropriate documentation is provided. The students should follow the submission procedure for each assignment. Failure to follow the procedure (e.g., incorrect, unreadable, and/or missing file attachments as instructed) heavily penalizes the assignment. Each assignment must be done *individually*. The last 7 calendar days prior to, and including, the last day of classes are free from any procedures for which a mark will be assigned.

4. **Midterm Exam:** Should a student miss a midterm exam, with appropriate documentation and verifiable reason, the weight of the missed midterm exam will be moved to the final exam. The results of the midterm exam will be released to students at least 2 days prior to the voluntary withdrawal deadline as per the [Senate Bylaw 54: Undergraduate Academic Evaluation Procedures](#).
5. **Final Exam:** Students who miss a final exam for a verifiable reason and with appropriate documentation will be given a make-up exam prior to the submission of final course grades that carries the same weight and measure the same knowledge.
6. **Make-up of the Make-up:** There will be *no* make-up of the make-up exam, and the final grade will be assigned based on overall work.

- 81
- 7. **Required Documentation for Missing Exam:** Medical or compassionate documents for the missing of an exam must be submitted within 1 week of the exam. Students are responsible for notifying the instructor that they will be missing an exam as soon as possible.
 - 8. **Accommodation for Religious or Spiritual Observance:** Requests for accommodation of specific religious or spiritual observance must be presented to the instructor no later than 2 weeks prior to the conflict in question (in the case of final examinations within two weeks of the release of the examination schedule). In extenuating circumstances, this deadline may be extended. If the dates are not known well in advance because they are linked to other conditions, requests should be submitted as soon as possible in advance of the required observance. Timely requests will prevent difficulties in arranging constructive accommodations.
 - 9. **Academic Accommodation:** A student who has 3 or more major in-term evaluations scheduled or due within 24 hours may apply, no later than the end of the first quarter of classes, to seek an appropriate accommodation such as a due date extension, alternative assignment, or rescheduled exam.

10. **Appeal:** Students have the right to review the exams and assignments marking within 1 week of their release.

83

83

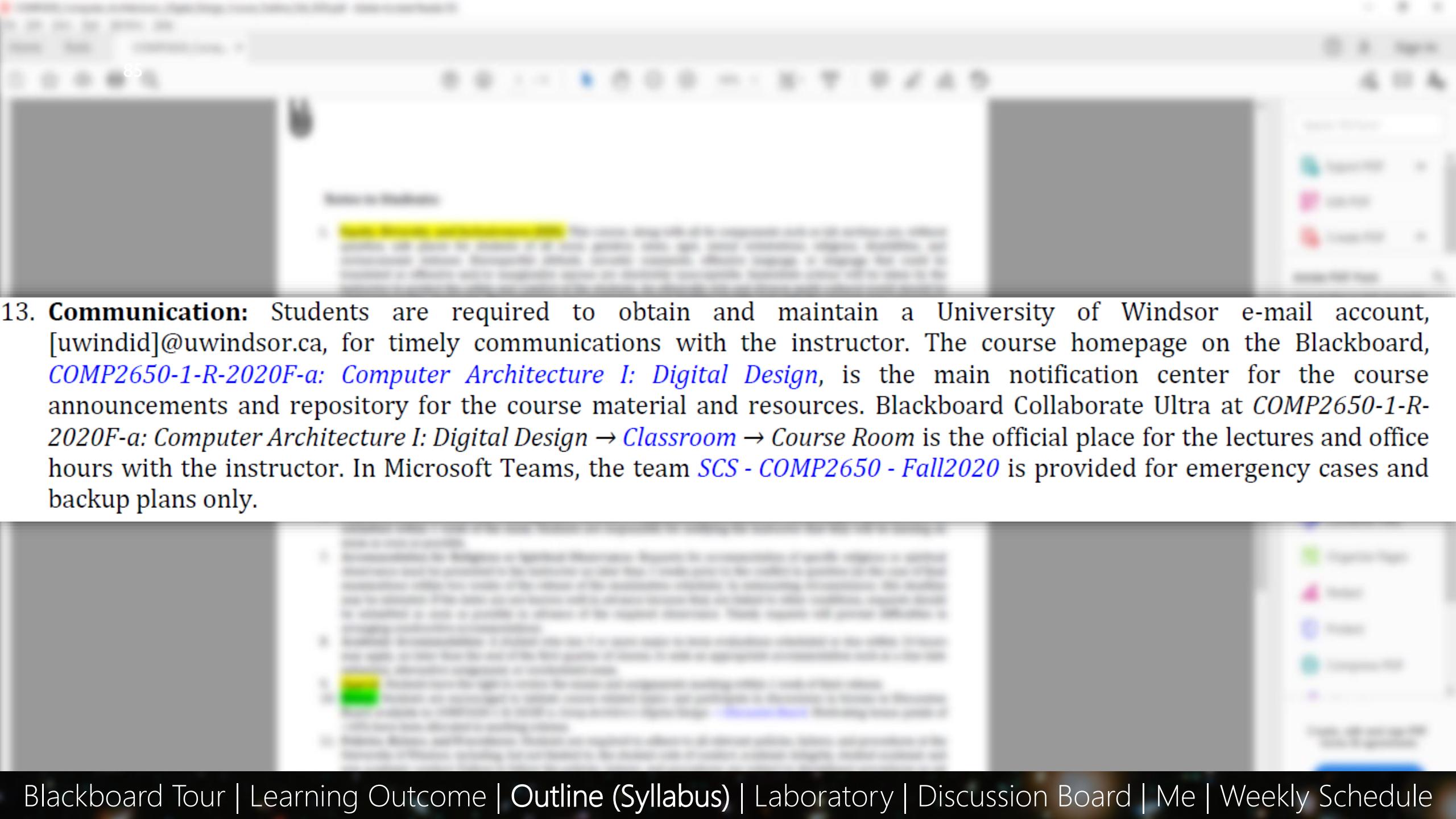
11. **Bonus:** Students are encouraged to initiate course-related topics and participate in discussions in forums in Discussion Board available in *COMP2650-1-R-2020F-a: Computer Architecture I: Digital Design* → *Discussion Board*. Motivating bonus points of +10% have been allocated in marking schema.

Blackboard Tour | Learning Outcome | Outline (Syllabus) | Laboratory | Discussion Board | Me | Weekly Schedule

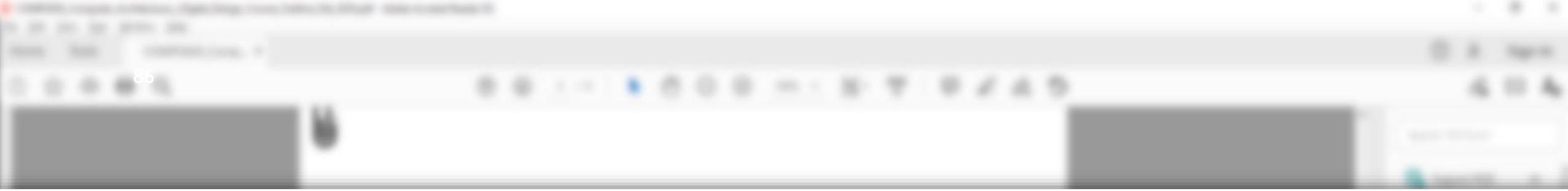
84

12. **Policies, Bylaws, and Procedures:** Students are required to adhere to all relevant policies, bylaws, and procedures at the University of Windsor, including, but not limited to, the student code of conduct, academic integrity, student academic and non-academic conduct. Failure to follow the policies, bylaws, and procedures are subject to disciplinary procedures as set out under, but not limited to, the [Senate Bylaw 31: Academic Integrity and Procedures for Addressing Student Non-Academic Misconduct](#). Regarding the plagiarism, the Blackboard's SafeAssign will be used for some or all student assignments or equivalent at the instructor's discretion. Plagiarized submissions or equivalent (e.g., exams), i.e., submissions with the same or minor modifications, receive zero. Should you need to record the lectures, please follow the [Senate Policy on Recording Lectures](#).

85



13. **Communication:** Students are required to obtain and maintain a University of Windsor e-mail account, [uwindid]@uwindsor.ca, for timely communications with the instructor. The course homepage on the Blackboard, *COMP2650-1-R-2020F-a: Computer Architecture I: Digital Design*, is the main notification center for the course announcements and repository for the course material and resources. Blackboard Collaborate Ultra at *COMP2650-1-R-2020F-a: Computer Architecture I: Digital Design* → *Classroom* → *Course Room* is the official place for the lectures and office hours with the instructor. In Microsoft Teams, the team *SCS - COMP2650 - Fall2020* is provided for emergency cases and backup plans only.



14. **Change Notification:** Any changes in the course outline, exam dates, marking, or evaluation will be discussed in class at least 2 weeks prior to being implemented.
15. **Student Evaluation of Teaching (SET):** The Student Evaluation of Teaching (SET) will be conducted during the last 2 weeks of the classes.
16. **Online Experience:** Participants in online lectures and lab sections include an instructor, a moderator, and students. Students are able to share camera or send messages but cannot share audio unless they Raise Hand, and the moderator or the instructor allows them. The moderator also supervises private messages. Students are encouraged to let the moderator and/or the instructor know of any connection issues asap regarding the quality of presentation in terms of audio and video (e.g., slides).

A blurred screenshot of a Blackboard discussion board. The board shows a list of posts from various users, each with a small profile icon and a timestamp. The posts are arranged in a grid-like structure with horizontal and vertical lines separating them.

17. **Feeling Overwhelmed?** Should face obstacles and experience difficulties that affect her academic performance, students can reach out to the following service centers as well as other on- and off-campus resources listed here www.uwindsor.ca/wellness:
- Student Health Services
 - Student Counselling Centre
 - Peer Support Centre

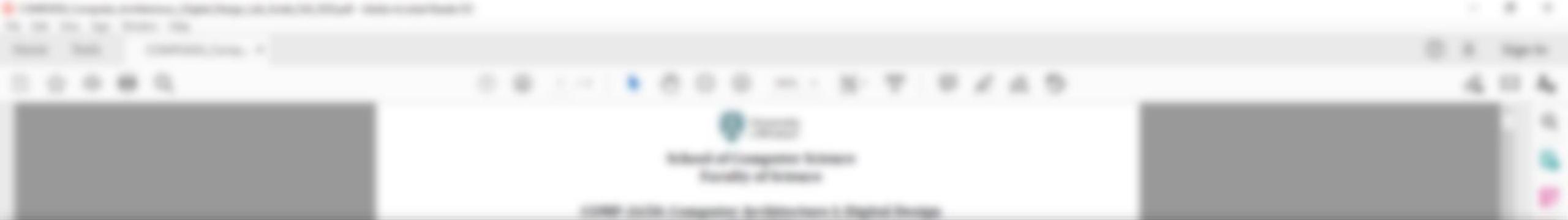
LABORATORY GUIDE

aka. Lab Manual

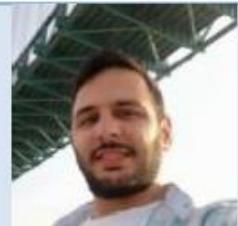
Labs → [COMP2650 Computer Architecture I Digital Design Lab Guide.pdf](#)

level 1.0

student work



Lab Instructors (GA) Lab Assistants (TA)



Ala Alam Falaki (GA)
alamfal@uwindsor.ca



Saiteja Danda (GA)
saitaja@uwindsor.ca



Shivani Pansara (TA)
pansara@uwindsor.ca



Ariya Rasekh (TA)
rasekh@uwindsor.ca



Phillip Pham (TA)
pham11l@uwindsor.ca



Husin Sarhill (GA)
sarhilla@uwindsor.ca



Mohamad Farhat
farha116@uwindsor.ca



Sean Janssens
janssen1@uwindsor.ca

Lab Sections

	blackboard.uwindsor.ca→COMP2650-1-R-2020F-a: Computer Architecture I: Digital Design→ Labroom Lab Instructor (GA): Saiteja Danda (saiteja@uwindsor.ca) ² Lab Moderator (GA): Ala Alam Falaki (alamfal@uwindsor.ca) Lab Assistant (TA): Ariya Rasekh (rasekh@uwindsor.ca) Lab Assistant (TA): Mohamad Farhat (farha116@uwindsor.ca) Section: 51, Mondays 11:30AM - 12:50PM, Blackboard Collaborate Ultra Section: 53, Mondays 01:00PM - 02:20PM, Blackboard Collaborate Ultra Section: 55, Tuesdays 11:30AM - 12:50PM, Blackboard Collaborate Ultra
	Lab Instructor (GA): Ala Alam Falaki (alamfal@uwindsor.ca) Lab Moderator (GA): Saiteja Danda (saiteja@uwindsor.ca) Lab Assistant (TA): Shivani Pansara (pansara@uwindsor.ca) Lab Assistant (TA): Husin Sarhill (sarhill@uwindsor.ca) Section: 52, Wednesdays 11:30AM - 12:50PM, Blackboard Collaborate Ultra Section: 54, Wednesdays 01:00PM - 02:20PM, Blackboard Collaborate Ultra Section: 55, Tuesdays 11:30AM - 12:50PM, Blackboard Collaborate Ultra

A blurred screenshot of a Blackboard course page. At the top, there's a navigation bar with various icons. Below it is a header section with the university logo and some text. The main content area shows a syllabus or course outline. On the right side, there's a vertical color-coded navigation bar with squares in red, green, blue, yellow, and purple.

# Labs	10
A blurred screenshot of a Blackboard course page. At the top, there's a navigation bar with various icons. Below it is a header section with the university logo and some text. The main content area shows a syllabus or course outline. On the right side, there's a vertical color-coded navigation bar with squares in red, green, blue, yellow, and purple.	

A blurred screenshot of a Blackboard course page. At the top, there's a navigation bar with various icons. Below it is a header section with the university logo and some text. The main content area shows a syllabus or course outline. On the right side, there's a vertical color-coded navigation bar with squares in red, green, blue, yellow, and purple.

Submission Due Date

Next Week Wednesdays Midnight Anywhere on Earth (AoE)

A blurred screenshot of a Blackboard course page. At the top, there's a navigation bar with various icons. Below it is a header section with the university logo and name. The main content area shows a syllabus document titled "Course Syllabus - Introduction to Regional Geology" dated "Fall 2010". A large blue rectangular box covers the majority of the page content.

Grade Release Date	1 Week after Submission Due Date
A blurred screenshot of a Blackboard course page. It features a green sticky note with text, several small profile pictures of students, and a "View Details" button.	

Marking Scheme

10 Lab Assignments × 2% Each = 20%

Attendance

Encouraged but not mandatory due to time zone accommodation in the COVID-19 era.

A blurred screenshot of a Blackboard course page. At the top, there's a navigation bar with various icons. Below it is a header section featuring the university logo and the text "SCHOOL OF MANAGEMENT STUDIES Faculty of Business". A blue horizontal bar spans across the middle of the page. The main content area is mostly illegible due to blurring but appears to contain course information.

Office Hours

N/A.

LABROOM

aka. Laboratory

Labs → [How to Attend Lab Sections](#)

DISCUSSION BOARD

Lectures → How To Find → [How to Participate in Discussion Board](#)

ME

hfani.myweb.cs.uwindsor.ca



WSDM 2016, 2nd Year PhD, Baker Beach, San Francisco, USA.



Hossein Fani, PhD

Assistant Professor, School of Computer Science, Faculty of Science, University of Windsor
Room 5111, Lambton Tower

hfani@uwindsor.ca; hossein.fani@uwindsor.ca
hfani.myweb.cs.uwindsor.ca

OFFICE

Lectures → How To Find → [How to Attend Instructor's Office Hours](#)

WEEKLY SCHEDULE

Weekly Schedule

Eastern Time	SUN	MON	TUES	WED	THURS	FRI	SAT
7:00 AM					Deadline for Lecture and Lab Assignments Submission		
8:00 AM							
8:10 AM							
8:20 AM							
8:30 AM		Lecture A Instructor: Hossein Moderator: Phillip		Lecture B Instructor: Hossein Moderator: Phillip			
8:40 AM							
8:50 AM							
9:00 AM							
9:10 AM							
9:20 AM							
9:30 AM							
9:40 AM							
9:50 AM							
10:00 AM		Office Hour A Instructor: Hossein Moderator: Phillip		Office Hour B Instructor: Hossein Moderator: Phillip			
10:10 AM							
10:20 AM							
10:30 AM							
10:40 AM							
10:50 AM							
11:00 AM							
11:10 AM							
11:20 AM							
11:30 AM		Lab Section 51 Instructor: Saiteja Moderator: Ala Marking: Ariya	Lab Section 55 Instructor: Saiteja, Ala Moderator: Ala, Saiteja Marking: Ariya, Shivani	Lab Section 52 Instructor: Ala Moderator: Saiteja Marking: Shivani			
11:40 AM							
11:50 AM							
12:00 PM							
12:10 PM							
12:20 PM							
12:30 PM							
12:40 PM							
12:50 PM							
1:00 PM		Lab Section 53 Instructor: Saiteja Moderator: Ala Marking: Ariya	Lab Section 54 Instructor: Ala Moderator: Saiteja Marking: Shivani				
1:10 PM							
1:20 PM							
1:30 PM							
1:40 PM							
1:50 PM							
2:00 PM							
2:10 PM							
2:20 PM							
2:30 PM							
2:40 PM							
2:50 PM							
3:00 PM							
3:10 PM							
3:20 PM							
3:30 PM							
3:40 PM							
3:50 PM							

Eastern Time	SUN	MON	TUES	WED
8:30 AM				
8:40 AM				
8:50 AM				
9:00 AM		Lecture A Instructor: Hossein		Lecture B Instructor: Hossein
9:10 AM		Moderator: Phillip, Sean		Moderator: Phillip, Sean
9:20 AM				
9:30 AM				
9:40 AM				
9:50 AM				
10:00 AM				
10:10 AM				
10:20 AM		Office Hour A Instructor: Hossein		Office Hour B Instructor: Hossein
10:30 AM		Moderator: Phillip, Sean		Moderator: Phillip, Sean
10:40 AM				
10:50 AM				
11:00 AM				

Eastern Time	SUN	MON	TUES	WED
11:30 AM				
11:40 AM				
11:50 AM		Lab Section 51 Instructor: Saiteja Moderator: Ala Marking: Ariya, Mohamad	Lab Section 55 Instructor: Saiteja, Ala Moderator: Ala, Saiteja Marking: Ariya, Mohamad, Shivani, Husin	Lab Section 52 Instructor: Ala Moderator: Saiteja Marking: Shivani, Husin
12:00 PM				
12:10 PM				
12:20 PM				
12:30 PM				
12:40 PM				
12:50 PM				
1:00 PM				
1:10 PM				
1:20 PM		Lab Section 53 Instructor: Saiteja Moderator: Ala Marking: Ariya, Mohamad		Lab Section 54 Instructor: Ala Moderator: Saiteja Marking: Shivani, Husin
1:30 PM				
1:40 PM				
1:50 PM				
2:00 PM				
2:10 PM				
2:20 PM				

Eastern Time	THURS
7:00 AM	Deadline for Lecture and Lab Assignments Submission

CODELINE

Lectures → W01: Meet & Greet, Number Systems I → Codeline

THANK YOU!



QUESTION