

# CS1002 Programming Fundamentals (CYSEC)

---

Fall 2022  
**23<sup>rd</sup> August 2022**

**Jawad Hassan**  
Lecture 1

National University of Computer and Emerging Sciences, Islamabad

---

# Welcome to Cyber Security

Welcome!

Plan your self for next four years with Cyber  
Security

---

# Ubiquity of Communication Devices

---



Figure 1: Computers are everywhere, processing and generating data.

---

# Internet of Things



Figure 2: Everything is connected and sharing data ...

# What is cyber Security?



### Figure 3: What is Cyber Security?

---

# Application of cyber security

# Man vs Machine

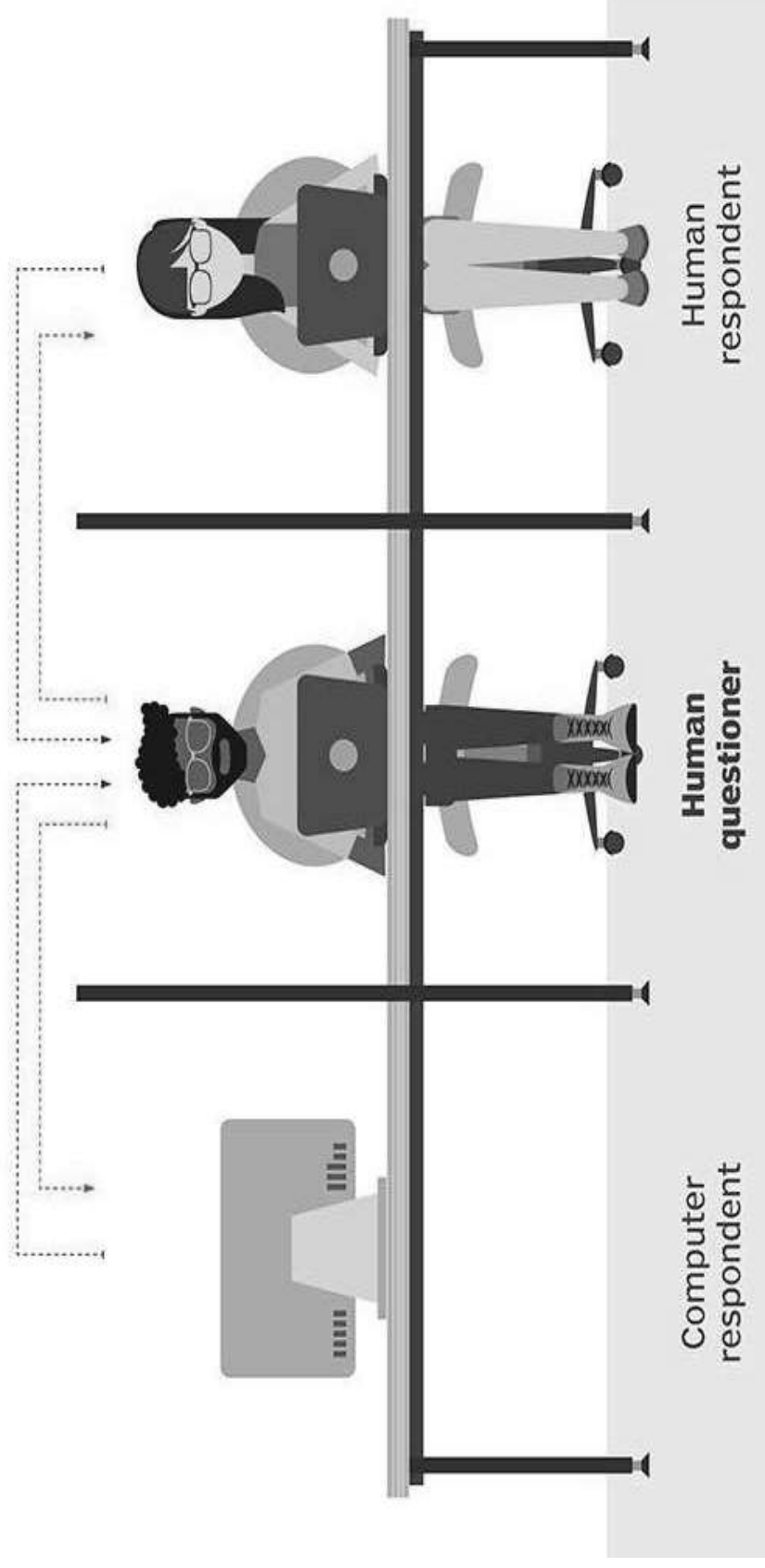


Figure 4: Man vs Machine

# Objectives of Programming Fundamentals

- Problem Abstraction
  - Problem understanding from a computational perspective
- Design
  - Developing a conceptual solution
- Implementation
  - Implementing the solution as a computer program
- Testing
  - Validating the implementation



---

# Programming is like Legos

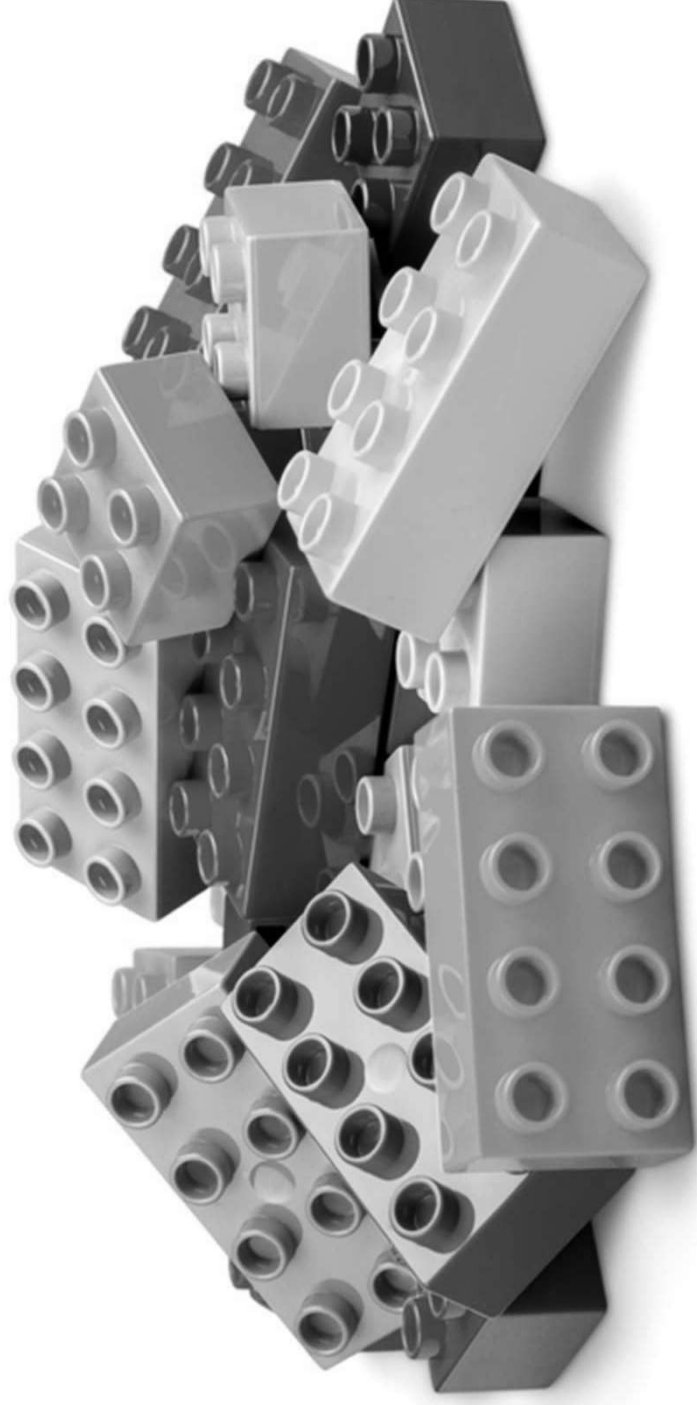


Figure 5: Small number of primitive constructs! [basic tools and pieces]

# Programming and Legos!

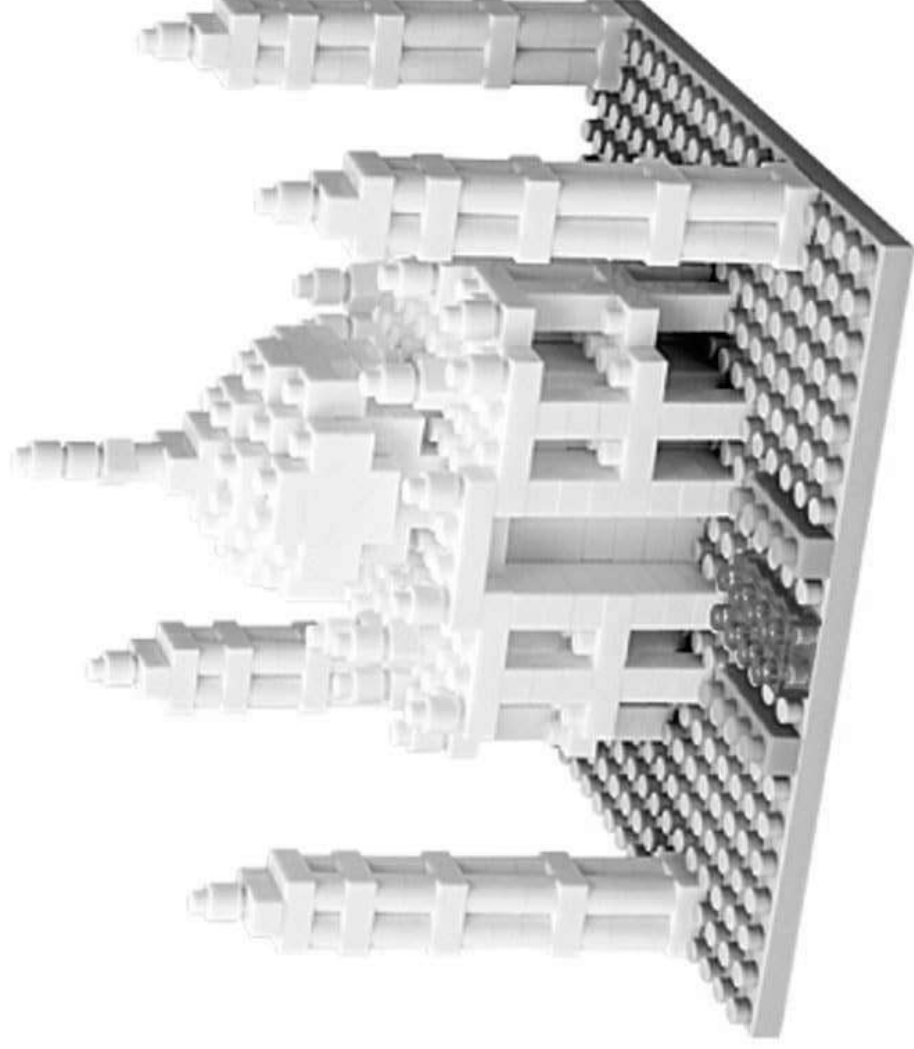


Figure 6: You can build huge, elaborate structures out of these pieces!

---

What is the most important skill of a Computer Scientist ?

- What is the most important skill for a computer scientist?
  - a) To think like a computer.
  - b) To be able to write code really well.
  - c) To be able to solve problems.
  - d) To be really good at math.

Why Google and why not Bing ?

---

---

# Skills of a Computer Scientist

- A computer scientist must have these three Critical Skills:
    1. Problem Solving
    2. Programming (master of communicating his solutions to computer)
    3. Strong Math/Physics Knowledge
-

---

# Our Objectives

- To teach you problem solving capabilities by teaching you how to think like a computer scientist.
- This involves teaching the complete steps from problem understanding to testing your solution.

---

# How you can become a good CS ?

- No Royal Way.
  - Dedication / Ambition
  - Hard Work
  - Punctuality
  - Professionalism (Integrity and Honesty).
- 
- **Simple Rule: Program 2-3 Hours a Day to Pass it [Learning]**

---

# What Subjects to Specially Focus on in CS?

- Problem Solving
  - Programming
  - Maths (every single subject).
-

- 
- # Skills Required for an Excellent Computer Scientist
- Computational Thinker
  - A Keen Mathematics (Linear Algebra, Probability and Calculus are extremely important)



---

# What to do for being Successful in CS1002

## Or In Semester System

- Be punctual
  - Do not put off your today work till tomorrow.
  - Start on the given task as soon as it is given.
  - Set your goals, for each day, week and month.
- Be attentive and proactive in the class.
- Communicate your problems and discuss them immediately. Don't be Shy and ask questions.
- Think big and do not limit yourself to the class content, **explore and imagine.**

# Advantages of doing good in CS1002

---

- Many
  - Will help you to appreciate your own mind power and act as master of computer.
  - Can help you to understand the other CS core courses thoroughly.
  - You can start working on real problems soon after completing it and CS1002.

---

# I Just Hate Teaching Passive Classes

---

- Be Proactive
  - Ask questions → Questions are gateway to your understanding
  - Stop me right there when you do not understand anything.
  - The more responsive you will be better you will learn.
- Rules For Class participation
  - No talking and laughing during a lecture
  - If you have any question or a comment/suggestion, raise your hand and only speak if you are given permission to do so.

---

# Class Discipline

- No use of mobile phones during a lecture
    - Mobiles must be switched off at the start of a lecture and must be put in a bag or pocket i.e. they should not be seen in anyone's hand or on desk.
  - No attendance for late comers (Be aware Attendance < 80 % == Debarred means no Final Exams)
    - Attendance will be taken at the start of a class only.
    - In the case of leaving during a lecture (due to any emergency), you will tell me so that you can be marked absent. Try to finish all things before coming to a class.
    - Moreover Quizzes will be mostly taken during first 10 minutes, so being late == losing marks as well.
  - Lecture Rooms
    - Section-C: Tuesday (C-409 8:30-09:50), Thursday (C-405 8:30-09:50)
    - Section-D Tuesday (B-227 10:00-11:20), Thursday (C-406 10:00-11:20)
    - Labs: Friday (Section-C 8:30-11:50 & Section-D 2:15-5:00)
-

---

# Credit Hours

- **Total credit hours = 4**
  - 3 credit hours of class lectures
  - 1 credit hour of Lab (equivalent to 3 hours of Lab work)

---

# Lab Work

- Lab Instructors are to be consulted for Lab related issues.
  - Do not switch lab sections without asking permission from the lab instructors.
  - Laboratory attendance will be taken in each lab.
  - For queries related to Lab exercises or assignments, consult the lab instructors only.
  - The rules related to the labs will be announced in the laboratory session.
-

---

# Grading Criteria

- Quizzes (5+2) = 10% → ~announced or un-announced
- Assignments (5) = 15 % → ~ Home works (Ungraded)
- Project (1) = 10%
- Sessional Exams (2) = 25 % (Paper-based exam)
- Final (1) = 40%

**Grading Policy: Absolute**

---

# Policy about missed assessment items in the course

- **Retake of missed assessment items** (other than **midterm/ final exam**) will not be held (**no retake of assignment/quiz/project**).

**Late submission will be accepted (until certain time) with marks deductions.**

- For a missed midterm/ final exam, an exam retake/ pretake application along with necessary evidence are required to be submitted to the department secretary. The examination assessment and retake committee decides the exam retake/ pretake cases.
-



---

# Course Plagiarism Policy

Plagiarism in any assessment item will result **zero marks** in that assessment.

Repeated case of plagiarism will be reported to the disciplinary committee and may result in zero marks **in the whole category**.

If plagiarism is detected, student will have **1 week** from the date of announcement to defend the charges

---

# Detailed Course Contents (1/2)

| List of Topics  | No. of Weeks | Contact Hours |
|---|--------------|---------------|
| <ul style="list-style-type: none"><li>- Problem-solving, Basic flowchart, block diagram, and programming languages.</li><li>- Primitive data types, input/output (hello world).</li><li>- Signed and unsigned data types, constants and variables.</li></ul>  | 1            | 3             |
| <ul style="list-style-type: none"><li>- Arithmetic operators (+, -, *, /, % and their compound counterparts) with their associativity and precedence.</li><li>- Bit wise operators</li></ul>  | 2            | 6             |
| <ul style="list-style-type: none"><li>- Function prototypes, definition, and calling.</li></ul>   | 1            | 3             |
| <ul style="list-style-type: none"><li>- Conditional/selection structures.</li><li>- Comparison and logical operators.</li><li>- if, if. . else and if else if structure.</li><li>- Switch statement, <i>break</i> statement.</li><li>- Ternary operator.</li></ul>  | 2            | 6             |
| <ul style="list-style-type: none"><li>- Repetition structures.</li><li>- Pre/post increment/decrement operators.</li><li>- while loop (sentinels + condition).</li><li>- Loop with <i>for</i>.</li><li>- Loop with <i>do-while</i>.</li><li>- Nesting of <i>while</i>, <i>for</i> loop and <i>continue</i> statement.</li></ul> | 3            | 9             |

# Detailed Course Contents (2/2)

|  |           |           |
|--|-----------|-----------|
| <ul style="list-style-type: none"><li>- Introduction to Arrays.</li><li>- Array initialization and representation.</li><li>- Char arrays.</li><li>- Multi-Dimensional Arrays (MDA).</li><li>- MDA representation in memory.</li></ul>                          | 1.33      | 4         |
| <ul style="list-style-type: none"><li>- Aliases, parameters passing by value and by reference (passing arrays).</li><li>- Function calling order and stack (function within a function).</li><li>- Recursion</li></ul>   | 1.66      | 5         |
| <ul style="list-style-type: none"><li>- Header files (creating own file).</li><li>- Files handling</li><li>- Opening flags (app mode).</li></ul>   | 1         | 3         |
| <ul style="list-style-type: none"><li>- Pointers.</li><li>- const. vs. non-const. pointers, a pointer to const. data vs. a pointer to non-constant data.</li><li>- Using pointers.</li><li>- Dynamic memory allocation.</li><li>- Array of pointers.</li></ul> | 2         | 6         |
| <b>Total</b>   | <b>15</b> | <b>45</b> |

---

# Our Team

## ■ Course Instructor

- Jawad Hassan
  - Section C and D
  - Office: 202-E
  - Email: [jawad.hassan@nu.edu.pk](mailto:jawad.hassan@nu.edu.pk)

## ■ Lab Instructors

- Ms. Sara Afzal ( 209-D A block)
  - Email: [sara.afzal@nu.edu.pk](mailto:sara.afzal@nu.edu.pk)
- Ms. Ayeha Qamar
  - Email: [ayesha.qamar@nu.edu.pk](mailto:ayesha.qamar@nu.edu.pk)

## ■ Course Coordinator

- Dr Mudassar Aslam
  - Section A and B
  - Office: 504e (5th Floor, Block C)
  - Email: [mudassar.aslam@nu.edu.pk](mailto:mudassar.aslam@nu.edu.pk)

---

# Course Coordination

## ■ Online course content & coordination

- ❑ Slate Course folder, flex for attendance and evaluations
- ❑ Google class room (<https://classroom.google.com>)
- ❑ Join class room by pressing **+**
- ❑ class code: **liimgp2**

# Course Consultation

## Consultation Hours:

| 🕒 Schedule and Office Hours (Fall-2022) |  |                      |                      |                      |                      |                      |
|---|--|----------------------|----------------------|----------------------|----------------------|----------------------|
| Days                                    | 8:30 AM<br>09:50 AM  | 10:00 AM<br>11:20 AM | 11:30 AM<br>12:50 AM | 13:00 AM<br>14:20 PM | 14:20 PM<br>15:50 PM | 15:55 PM<br>16:50 PM |
| Monday                                  |  |                      | Office<br>Hour       | Office<br>Hour       |                      |                      |
| Tuesday                                 | PF(CY-C)   | PF(CY-C)             |                      | Number<br>Theory     |                      |                      |
| Wednesday                               |  | Office<br>Hour       | Office<br>Hour       | Office<br>Hour       |                      |                      |
| Thursday                                | PF(CY-C)   | PF(CY-C)             |                      | Number<br>Theory     |                      |                      |
| Friday                                  | Appointment via Email ( <a href="mailto:jawad.hassan@nu.edu.pk">jawad.hassan@nu.edu.pk</a> ) |                      |                      |                      |                      |                      |

Helping hours (HH)

---

# Course Books

## ■ Text Books

- ❑ Starting Out With CPP ( 8<sup>th</sup> or 9<sup>th</sup> Edition) By Tonny Gaddis (Locally Available)
- ❑ [How to think like Computer Scientist, Interactive Edition](#) [Freely available online; Excellent Online Resource for Interactive Learning]

## ■ Reference Books

- ❑ C++ How to Program by Deitel & Deitel (8<sup>th</sup> Edition)
- ❑ C++ Without Fear : A Beginner's Guide that Makes you Feel Smart (English) 2nd Edition
- ❑ Accelerated C++.
- ❑ C++ Premier, 5<sup>th</sup> Edition

---

# Softwares

- Ubuntu (or any other recent Linux version) as operating system.
  - I would recommend installing Ubuntu LTS 18.04 or 19.04
    - Make copy of your important data before installation.
    - Read the instructions for steps to perform after the installation
- GNU compiler suite



---

# Live Programming & Visualization Environments

- **Blockly Games**, <https://blockly-games.appspot.com/> **or Snap** <http://snap.berkeley.edu/snapsource/snap.html>
- **C++ Live Program Visualization**
  - <http://cpp.sh/>
  - <https://www.jdoodle.com/online-compiler-c++>
  - [https://www.tutorialspoint.com/compile\\_cpp\\_online.php](https://www.tutorialspoint.com/compile_cpp_online.php)
  - [http://rextester.com/l/cpp\\_online\\_compiler\\_visual](http://rextester.com/l/cpp_online_compiler_visual)

---

# Tips and Advices

- **Self belief**
- **Honesty**
- **Respect**
- **Annual vs Semester System**
- **Attendance**
- **Deadlines**
- **Be interactive**
- **Make good friends**
- **Time table**

# Assignment 0 (No credit assignment)

- Task 1
  - Install Ubuntu on your machine or in a VM
  - Download Desktop Ver 22.04 LTS or any other LTS version (<https://ubuntu.com/blog/tag/LTS>)
  - Install GNU Compiler Collection (GCC) on Ubuntu
- Task 2
  - Time table of semester
- Task 3
  - Join Google Class Room
  - class code: [liimgp2](#)

---

# Play Some Games

- <https://blockly-games.appspot.com/>
- <https://scratch.mit.edu/projects/editor/?tutorial=started>

---

# QUESTIONS

---

---

**THANK YOU ALL**

---