



# CS1002 – Programming Fundamentals

Lecture # 02  
Tuesday, August 23, 2022  
FALL 2022  
FAST – NUCES, Faisalabad Campus

Muhammad Yousaf



# Algorithms and Problem Solving Techniques



# Basic Concepts of Problem Solving

- **Computer programming** is traditionally thought of as problem-solving process.
  - A **computer program** is a solution to a problem
- **Computer program:** The **computer program** is a set of instructions that tells the computer hardware what to do and when to do it
  - There are two parts to develop computer programs, **the algorithm** and **the syntax**
- **Algorithm:** The program's algorithm is the development of step-by-step, logical process that the program will follow to reach the desired goal of the program (the solution)
- **Syntax :** It is the rule of the programming language which dictate proper statement structure and usage



# Five Basic Elements of Computer Programming

- **Input:** Getting data and commands into the computer
- **Output:** Getting results out of computer
- **Conditions and repetition:** Decisions and cycling through instructions until some conditions if met
- **Mathematical operations:** Perform mathematical operations on the data
- **Variable and data structures:** Storing data that may change over time
  
- E.g. Vending machine



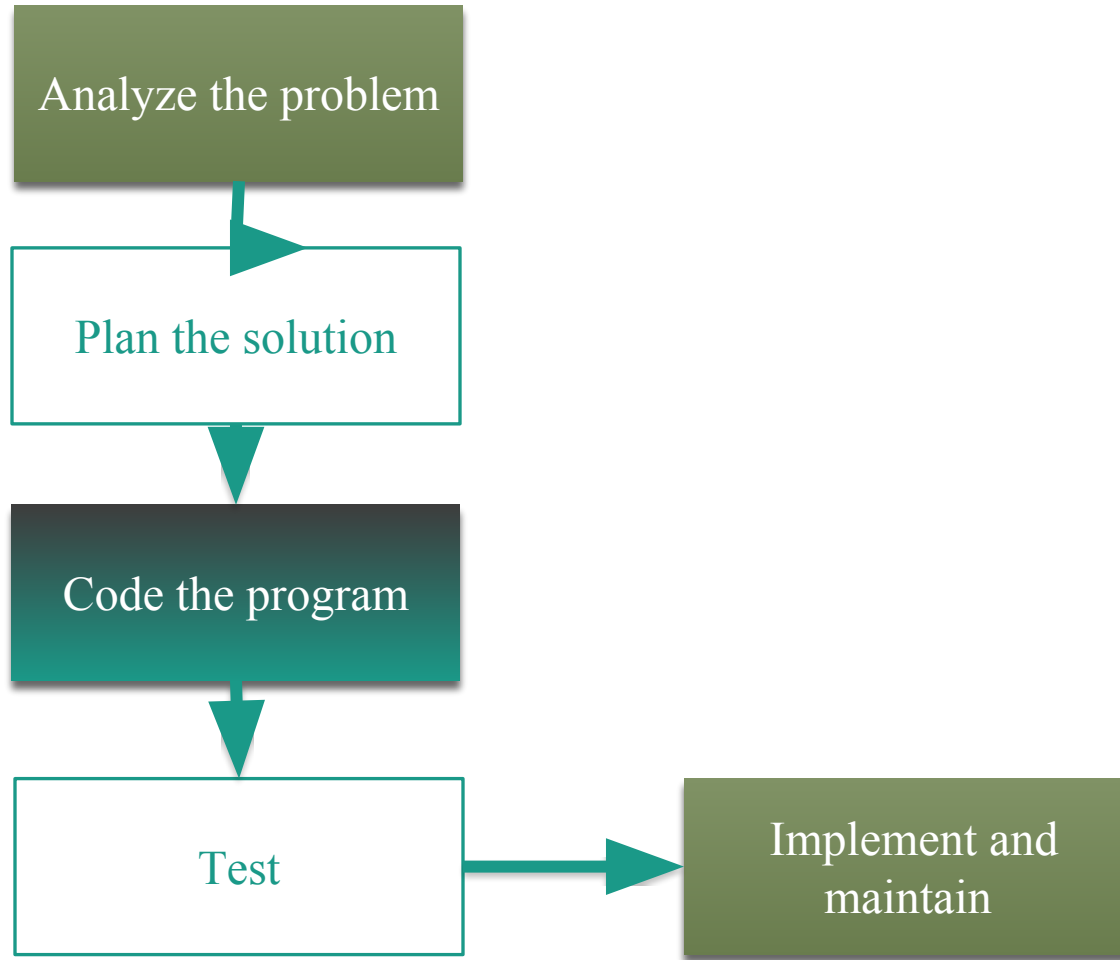
# The problem-solving process

There are several steps in the **program development cycle**

**Program development cycle consists of five steps**

- Analyze the Problem
- Plan the Solution
- Code the Program
- Test
- Implementation/Deployment and maintain

# The problem-solving process ...





# The problem-solving process

## 1. Analyze the problem : Programmer determines

- What the **program is supposed to do** (the purpose of the program)
- What **data** the program will use ( the programs input);
- What the program is supposed to **produce** (the programs output);
- The **process** the program will use to transform the data into the desired **output information**
- Visual tools for this stage can include
  - **Input layout charts** that represent how the input data is stored in the records of a file or how input prompt will appear on the screen
  - **Output layout spacing charts** represent what the finished output report or screen will look like



# The problem-solving process ...

2. **Plan the solution** : During the planning stage of the programming development cycle, the programmer utilizes visual tools such as flow charts, pseudocode and hierarchy charts to develop the programs **algorithm**, or solution to the problem

- A **flow chart** is a graphical representation step-by-step that shows the flow of control of the program using symbolic diagrams.
- **Pseudocode** is a visual representation of the same step-by-step logic, but pseudocode is English-like phrases instead of symbols
- **Hierarchy (or structure) charts** show the major operational tasks required for the program solution. In addition, the charts demonstrate the relation ships between each of the major sections
- Once the algorithm has been documented using one or more of the visual tools, the programmer checks the programs **logic** by stepping through the algorithm with realistic **test data**
- At this point the logic errors may be detected and corrected





# The problem-solving process ...

3. **Code the program:** The program (or source) code is written in the programming language selected by the programmer following the rules (or syntax) for that language
  - Once the source code is written
  - The program is processed by the **language translator** program
  - Any **syntax errors** are detected by the translator must be corrected before the machine language (object) can be generated
  - When all debugging and syntax errors is complete, a run-time version of the program can be executed
  - **Language translator:** A language translator converts **human-readable source code statements** into the **machine- readable object code**; depending on the language, the translator will be an **assembler**, **interpreter**, or **compiler** program



# The problem-solving process ...

4. **Test:** Testing the program is done using sets of data designed to produce the expected results
- If the program is faulty, the desired results will not be produced
  - The programmer must then **debug** the source code and revise the logic in the **planning stages**
  - This stage should require **minimal effort**
  - **Debug and revise:** To debug a program, the programmer finds and corrects syntax errors in the source code



# The problem-solving process ...

5. **Implementation/Deployment and maintain:** The final stages of the programming process is to put the program into production
- At this stage, all program documentation must be completed and presented at the time the program is implemented
  - The documentation includes all the documents used in the **planning stage** (such as input, output charts). The printed source code also becomes a part of the documentation
  - In addition, user **training manuals** are provided as well as any other information that the end user might require to properly run the program
  - **Maintaining** the program includes making appropriate updates to the program as needed
    - For instance, if income tax rates change, an update to the tax amounts would be required for a payroll program



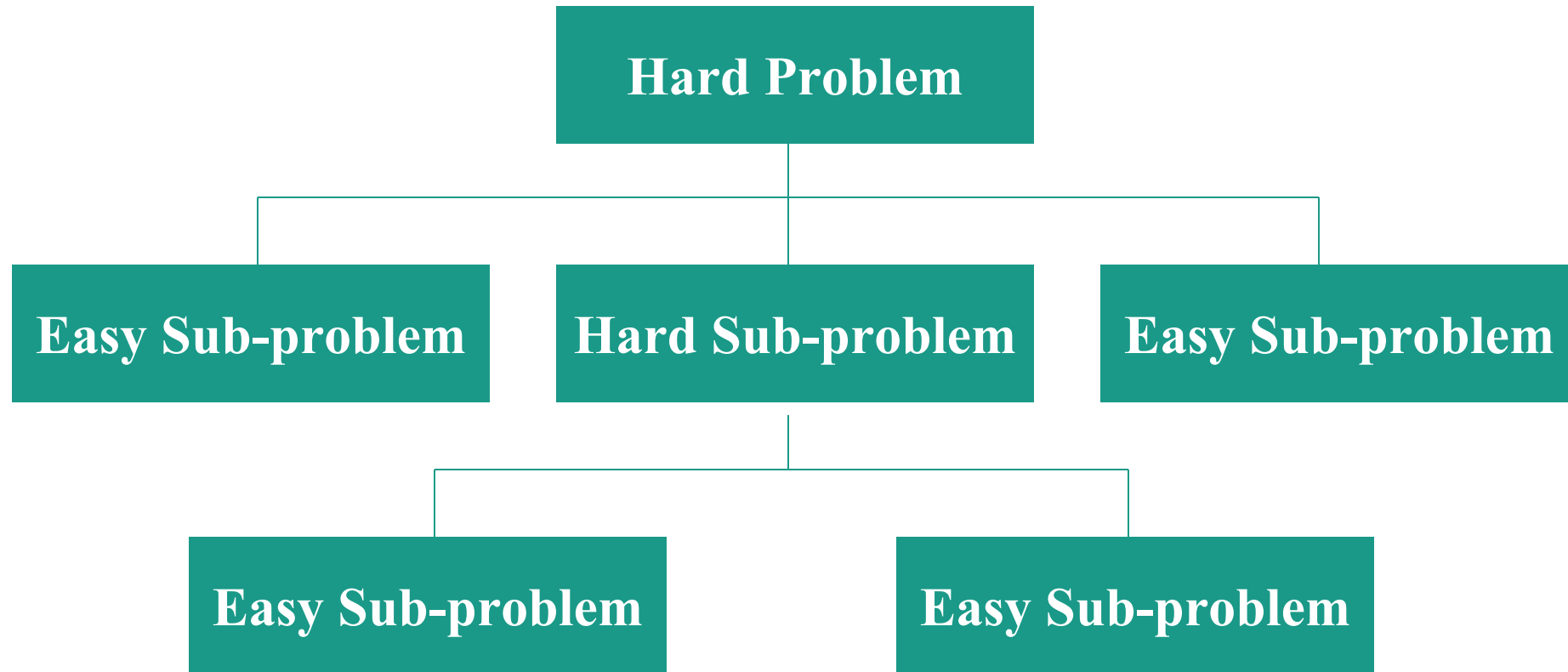
# Algorithms and Flowcharts



# Al-Khwarizimi Principle

- All complex problems can be broken into simpler sub-problems
- Solve a complex problem by breaking it down into smaller sub-problems and then solve them (in a specified order), one at a time
- When all the steps are solved, the original problem itself has also been solved
- This process is called Algorithm

# Divide and Conquer





# Algorithms

- A concept that pervades all areas of computer science
- Algorithm is a process that a computer could carry out to complete a **well defined task within finite time and resources**
- The **objective of computer science** is to solve problems by developing, analyzing, and implementing algorithmic solutions



# Steps in Problem Solving

- First produce a general algorithm (one can use **pseudocode**)
- Refine the algorithm successively to get step by step detailed **algorithm** that is very close to a computer language
- **Pseudocode** is an artificial and informal language that helps programmers develop algorithms
  - Pseudocode is very similar to everyday English





# Algorithms & Pseudocode

- A typical programming task can be divided into two phases
- **Problem Solving phase:**
  - Produce an ordered sequence of steps that describe solution of problem
  - This sequence of steps is called an **algorithm**
- **Implementation phase:**
  - Implement the program in some programming language



# Sample problem

- Input two numbers from the user and print the sum of those two numbers?

## Analyze the problem:

- What are the inputs?
  - Two number (Where to store these)
    - Need two containers
- What is the output?
  - A result of sum of the two numbers (Where to store it)
    - Need another container
- What is the process?
  - Perform the arithmetic operation of summation



# Sample problem

Plan the solution:

Steps Involved (Algorithm)

1. Take input of one number and store in a container named as **num1**
2. Take input of another number and store in another container named as **num2**
3. Perform operation **num1 + num2** and store the result in another container names **results**
4. Print the value in **results** on the screen



# Sample problem (Pseudo code)

1. declare num1, num2, results
2. input num1
3. input num2
4. results  $\square$  num1 + num2
5. Print results

# Questions

