CS6102– Computational Thinking

Week - 6

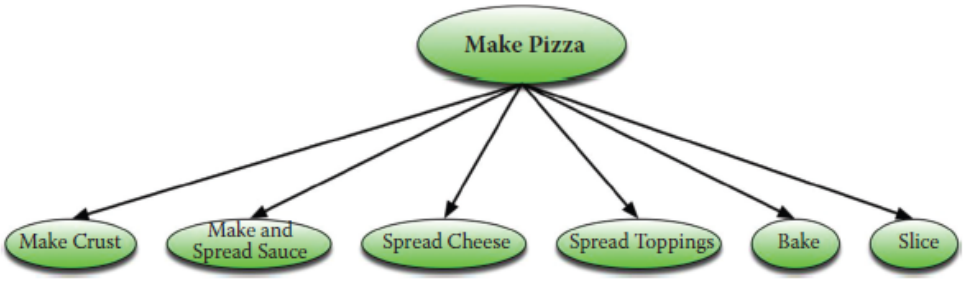
21 – Dec - 2022

Decomposition, Pattern Recognition, Abstraction, Generalization

Decomposition

is an approach that seeks to break a complex problem down into simpler parts that are easier to deal with. Its particular importance to CT comes from the experiences of computer science. Programmers and computer scientists usually deal with large, complex problems that feature multiple interrelated parts. Decomposition is a **divide-and-conquer** strategy. Simply put, it is the process of breaking down complexity by dividing a problem into sub problems. Visually this gives the problem definition a **tree structure**. Some terms related to Decomposition: Multitasking, Parallel programming, top-down design, successive prototyping, grid computing.

The motivation is that, solving a series of simple problems is easier than trying to tackle a big complex one.



TREE STRUCTURE

I. Main idea
    A. Sub-point of idea I.
        1. Sub-point of I.A.
        2. Sub-point of I.A.
    B. Sub-point of I
        1. Sub-point of I.B.
        2. Sub-point of I.B.
        3. Sub-point of I.B.
            a) Sub-point of I.B.3.
            b) Sub-point of I.B.3.
II. Main idea
    A. Sub-point of idea II.
    B. Sub-point of idea II.
III. Main idea

OUTLINE WAY

Pattern Recognition

Once a problem has been decomposed into smaller problems we can look for similarities or patterns. These patterns can help us solve problems correctly, quickly and efficiently.

"Pattern recognition is about spotting if any steps in a solution can be repeated. It can also help to think about whether we have solved similar problems before".

Recognition of similar characteristics

Example:

Consider drawing three different bicycles. They have some common features ike handle, break, two wheels, seat etc.,

- by recognising this pattern for all bicycles we can draw the first cycle and then draw two more that look similar.
- The only bits that will change in the drawings will be the specific details -
    - They could have different colour
    - They could have different back carriages
    - They could have a gear or no gear
    - They could be different sizes

Why Pattern Recognition?

- Patterns make our task simpler.
- Problems are easier to solve when they share patterns, because we can use the same problem-solving solution wherever the pattern exists.
- The more patterns we can find, the easier and quicker our overall task of problem solving will be.

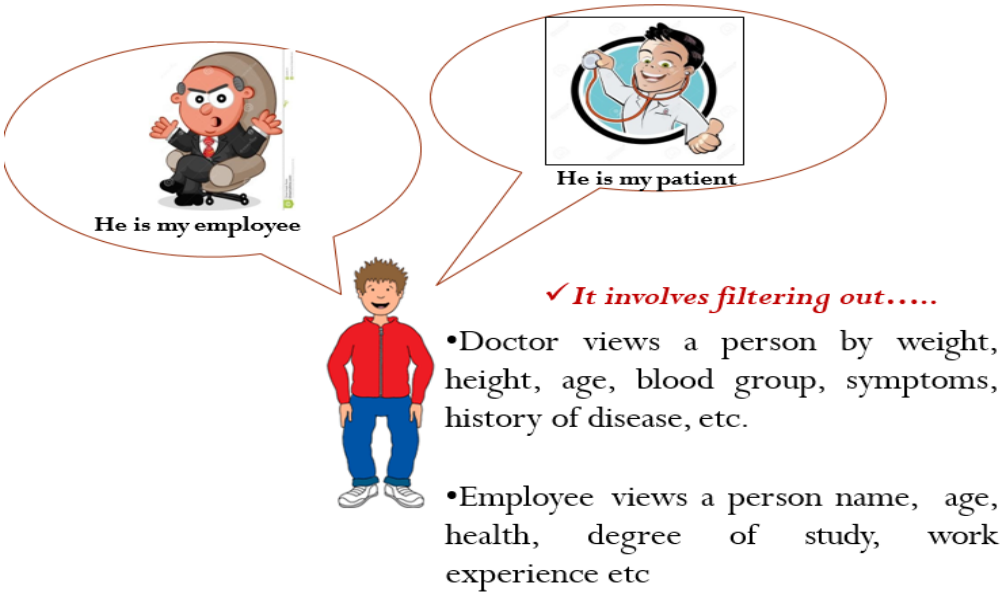
Abstraction

Abstraction:

- *Is a way of expressing an idea in a specific context while at the same time suppressing details irrelevant in that context.*
- *of something operates at a certain level of detail and puts a layer over the top to obscure some of the information.*

Some related terms: Modelling, Layers, modularity, prototyping, conceptualization, generalization, hiding insignificant details

- Once patterns are recognised, abstraction is used to gather the general characteristics and to filter out of the details we do not need in order to solve our problem.
- It involves filtering out – essentially, ignoring - the characteristics that we don't need in order to concentrate on those that we do.
- It filters the specific details.



Example :

- Consider the activity of drawing a series of cats.
- We noted that all cats have general characteristics, which are common to all cats, eg: eyes, a tail, fur, a liking for fish and the ability to make meowing sounds.
- In addition, each cat has **specific characteristics**, such as **black** fur, a **long** tail, **green** eyes, a love of **salmon**, and a **loud** meow.
- **These details are known as specifics**.
- We **don't** need to know what sound a cat makes or that it likes fish. These characteristics are irrelevant and can be filtered out.
- We **do** need to know that a cat has a tail, fur and eyes, but we **don't** need to know what size and colour
- Abstraction is gathering of *general characteristics* we need and *filtering out* of details and characteristics that we do not need.

Generalization

Generalisation is a way of quickly solving new problems based on previous problems we have solved. We can take an algorithm that solves some specific problem and adapt it so that it solves a whole class of similar problems. Then whenever we have to solve a new problem of that kind we just apply this general solution.

- Separate, but similar concepts, can be generalised into a single concept.
- Reuse it in other situations and solutions

Ex: ‘shape’ is a generalisation of line and circle.

EXERCISES on DECOMPOSITION, PATTERNS

Qn1. Decompose the given tasks into smaller tasks.

Note : Use either an OUTLINE way (a bullet point for each smaller steps) or a TREE structure or BOTH. (Attempt any 4 only)

a) Understanding how a Bicycle works
b) preparing for a weeklong trip
c) Searching for a book in a physical library
d) preparing a recipe of your choice for an event (in 3 possible ways one to satisfy the children, young adults, old)
e) identifying a region of the state from spoken languge
f) computing a simple quadratic equation / adding all numbers from 1 to 1000

From the decomposed smaller tasks, identify the patterns/similarities

Qn2. Identify few common features and different features for making a recipe in 3 different ways.

Qn 3: Decompose the drawing of this simley emoji. Identify the common patterns. Also list down the general and specific details pertaining to it. If the mouth is modified to represent other emtions.

Activity 1:

Assume there is a team of 10 people. If the team is asked to search for a toy kept hidden in any of the rooms in the department, design a strategy to search for the toy.
Will the same strategy be a good one if there are 20 people in the team? Can you think of a better strategy?
Will the same strategy be a good one if there are 4 people in the team? Can you think of a better strategy?

EXERCISES on ABSTRACTION, PATTERNS

Activity-1:
Order the following into l*ayers of abstraction*, starting with the most *general* and ending with the most *specific*.

- Peacock;
- bird;
- sweety the Peacock;
- animal;
- Indian Peacock.

Activity-2: Give your comments on the details and use of a *Map* of a place / region.

Activity-3:

From the points below identify the patterns and abstract:

- Cricket is a bat-and-ball game played between two teams of eleven players
- cricket can also bring other benefits and opportunities such as: Team skills. Social skills such as cooperation, communication and learning how to cope with winning and losing.
- Cricket is *a game of strategy between two teams*.
- Forms of cricket range from Twenty20, with each team batting for a single innings of 20 overs.
- Cricket is the second most popular sport in the world.
- There are **three** formats of cricket played at the international level – Test matches, One-Day Internationals and Twenty20 Internationals.

OBJECT DESCRIPTION - ABSTRACTION and MODELING

Describe an object that is used every day. Imagine that the object has j**ust been invented** and has to describe the object.

1. Think about and list the attributes of the object that are needed to describe and define the object
2. List the behaviors of the object.
    1. The functions / uses of the object
    2. What are the needs fulfilled by the object?
    3. The physical attributes and characteristics of the object (components or parts, shape or materials, general dimensions, connections between parts
3. There are some specifications to describe all of the attributes and functions of the object:

    1. Use clear, non-technical language, to describe the object's function, the need it fulfills and its attributes.
    2. Your description must be specific enough so that someone who has never seen the object could visualize it, understand how it works, and appreciate the benefits it provides.
    3. Describe the object using at least 150 words and listing a minimum of 6 attributes. Keep in mind that attributes should involve all of your senses. (e.g. Is it smooth? Does it make a noise? Does it have an odour? etc)

Objects that can be considered:

- **Scissors**
- **Measuring tape**
- **Stapler**
- **Umbrella**
- **Scale**
- **Key**
- **Notebook**
- **Television**
- **Mobile phone**
- **Knife**
- **Spoon**
- **Tumbler**
- **Pen**
- **Printer**

Mapping to Computer Science:

Q1. Consider your object as if you were a computer program. Let's draw a diagram that shows all of its functions as boxes, and for each function, its inputs (i.e. what is required to use the object in this way?) and outputs (i.e. what is produced when the object is used in this way?). Now draw the diagram of the object. Ask, Are there inputs and outputs that repeat for different functions?

Q2. Are there functions that are similar and can be combined so that the object can be represented with a more concise program (i.e. fewer steps or boxes)?

Q3. Think about the physical attributes and characteristics of your object. Organize these so that each is declared as a variable with its proper type (e.g. 'color,' 'material type,' 'height,' 'weight')**. Can some of these attributes and characteristics be arranged into a hierarchy of related attributes and characteristics** (e.g. 'Material Type' can indicate weight or color, but weight and color do not indicate material type)**?**

Q4. How does abstraction in Computer Science relate to the process of identifying the functions and characteristics as you have done in this exercise?

SOME MORE NOTES FOR REFERENCE:

Patterns are things that are the same within a problem and between problems.

- Identifying patterns means that there is probably an existing solution already out there.
- Pattern recognition is based on the 5 key steps of:
    - Identifying common elements in problems or systems
    - Identifying and Interpreting common differences in problems or systems
    - Identifying individual elements within problems
    - Describing patterns that have been identified
    - Making predictions based on identified patterns.
- Pattern abstraction is hiding the complexities of one pattern from another.
- Pattern generalisation is spotting things that are common between patterns.
- We can represent parts of a system in general terms, including Variables, Constants, Key Processes, repeated Processes, Inputs and Outputs.