

# Computational Thinking Concepts in IT 130

## **Concept 1: Tree Structure**

Trees are used as a structure for hierarchical data organization. The big idea is: organize large amounts of items into groups and continue partitioning each group into subgroups, recursively. A tree data structure allows efficient insertion and search. Understanding and using a tree data structure is a computational thinking instance that belongs in the Recollection category.<sup>1</sup>

**Learning Goal: Students recognize the application of a tree structure to organize data; they are able to take advantage of the structure to efficiently locate/reference data in the tree.**

### **Case 1: File System Structure/URLs**

URLs consist, basically, of a hostname and a pathname. The pathname is just the file system (relative) pathname corresponding to the file with the content. The file system pathname uses the hierarchical terminology to reference a file on a computer.

Discussion Questions:

1. Do you know anyone with 50 files on their desktop? How easy is it to find a file with this organization?
2. How are folders useful for organizing and finding files?
3. What's a good number of folders? If we have a large number of files, then we might end up with too many folders... How do we continue organizing the folders?
4. If I have 1000 files and 5 folders at each level. How many levels do I need?
5. How do we search for a file in a file system?
6. How can we represent a file location as text? How do URLs do this?

### **Case 2: Site Maps**

A web site is often organized as a tree. Web sites exist to offer content that can be partitioned into groups and, recursively, into subgroups. Again, this organization is to help users find the content they are looking for.

Discussion Questions:

1. How does a tree structure aid users in finding needed content on a web site?
2. How are web site categories organized within a tree?
3. How is navigating through a web site like searching through a tree?

---

<sup>1</sup> The categories we have currently are: Automation, Communication, Computation, Coordination, Design, Recollection, and Simulation.

4. Examine the web site for your college or university. Discuss how pages and links match a link structure? Are there links that do not match a tree structure?

### **Case 3: Domain Name Servers**

The Domain Name System is a computer application that connects many servers around the globe to provide the following service: translating a hostname to an IP address. (The destination host IP addresses is required to send information to it.) Domain name system is hierarchically organized to allow for efficient search of the DNS server that can provide a current IP number for a given a hostname.

Discussion Questions:

1. Why have IP addresses? Why have hostnames?
2. How does the domain name system find an IP number?
3. How does the hierarchical structure of the Domain Name System reduce the number of DNS queries for finding an IP number?
4. How does this hierarchical structure allow for distributed managing of the hostname/IP address pairs?

**Assessment: see Concept 2 Assessment.**

### ***Concept 2: Relative vs. Absolute References***

A setting or a location can be specified relative to a current state or in terms of an absolute specification. For example, a location of a building can be specified relative to a current location (e.g. 3 blocks north of here) or by an absolute location (e.g. 25 S. Elm St.). The concept of relative and absolute references are commonly used in IT applications and include file name references, cell references in a spreadsheet and configuration settings, such as those used in page formatting. Being able to understand and use absolute and relative references is an instance of computational thinking that belongs in the Recollection category.

**Learning goal: Students can find the relative reference given an absolute reference and state; vice versa, they can reconstruct an absolute reference based on a state and a relative reference.**

### **Case 1: Referencing URL/File location**

Items in a tree structure can be referenced through an absolute pathname, i.e. the sequence of groups containing the item, from largest to smallest. In many cases, these structures can be traversed and a relative pathname to an item is defined from the current group (containing the item).

Discussion Questions:

1. Given a web page and a hyperlink to another web page at the same web site, describe absolute and relative references

2. Why use relative references? How is a relative hyperlink reference useful if you transfer your files to a web server? What if you had used an absolute reference and then moved your files?
3. Consider a file finder, which user actions are absolute references? Which are relative to current location?

## **Case 2: Style Property Settings**

Many style properties can be specified by an absolute setting or relative to the current setting. For example, a margin setting may be given in terms of an absolute amount (1 inch margins) or relative to the current setting (increase current margin setting by 5em).

Discussion Question:

1. When is it useful to specify a relative setting? (Indentation for margin settings.)

**Assessment (also for Concept 1): Students will be given a visual representation of a simple file system tree and will answer several questions:**

- **Give the absolute pathname of file X**
- **Give the pathname of file Z relative to ancestor folder Y**
- **If moving folder D into folder C, list the files that will as well.**

## ***Concept 3: Class Membership***

A class is the abstraction of key properties of objects and patterns. It is used to hide less relevant or variable properties of the object or pattern, and to provide a way to create multiple instances (of objects or patterns) sharing the same properties. The ability to identify and abstract certain properties of a group of items into a class is an example of computational thinking that belongs to the Design category.

**Learning goal: Students can identify a group of (e.g. visual) elements that should all have the same property and abstract them into a class; they should then be able to specify class-wide (e.g. visual) properties that achieves the desired uniformity of the visual elements.**

## **Case 1: Style classes**

A style rule (using CSS) may specify visual formatting properties for a class. These formatting properties are then applied to all HTML tags that have the same class name as the style rule.

Discussion questions:

1. How does the use of a style class promote a consistent design across multiple web pages?
2. Can the class name be any sequence of characters? What are the advantages of giving it a meaningful name?

**Assessment: The students will write a CSS rule that will format all HTML tags of a specific class so that their content appears in a particular color.**

### ***Concept 4: Caching***

Caching replicates data originating from a remote resource locally, so it is close to the consumer of the data. The goal of caching is to improve data access latencies. What data is cached will depend on assumptions about future data accesses, which are usually not deterministic. Understanding the usefulness of caching and the ability to reason about what data to cache is example of computational thinking that belongs in the Reflection category.

**Learning Goal: Students understand the purpose of caching; they understand the issue of limited cache capacity and that the choice of what to cache is based on assumptions about future data access; they understand that cached data may be stale and will need to be re-synched with the original data.**

### **Case 1: Web Browser Caching**

Web browsers cache recently visited web pages to improve response times. The assumption is that future web page request will match recent web pages requests.

Discussion Question:

1. What web sites do you get back to quite often?
2. Do you mind waiting for a web page to load?
3. How can you minimize waiting for web pages you visit often?
4. Your solution raises issues of replica consistency... How do we deal with it?

### **Case 2: DNS caching**

DNS server cache recently obtained hostname - IP address pairs to improve response times. The assumption is again that future DNS requests will match recent ones.

Discussion Question:

1. How many DNS servers need to be queried in order to obtain an IP address for a web host in Sweden?
2. What penalty do we pay when we move from web page to web page on this Swedish host?
3. How do we using the caching concept to shorten the latency of web page downloads from this web site?

**Assessment: Student will be given the example of phone number caching on cell phones and will be asked the following questions:**

- **Why do cell phones cache recently dialed or received numbers?**
- **What limits the cache capacity? Is it just memory capacity or other factors?**
- **Give a reasonable argument for what numbers should be cached. Most recent? Dialed most often? Both?**
- **Is there an issue of cached data staleness in this example?**

### ***Concept 5: Communication protocols***

When two entities communicate, and particularly if they do so in an automated way, they follow a protocol that specifies, at any give step, the state of each entity (and, in particular, who is listening/receiving and who is talking/sending a message), messages that can be received/sent, and a state transition function that maps (state, received message) pairs to (new state, sent message) pairs. A protocol requires that at least one entity is a server that is initially in the listening state. Understanding the need for precise protocols for automated communication and understanding the (message) event-based state transition is an example of computational thinking that belongs to the Communication and Coordination categories.

**Learning Goal: Students understand that automated communication requires a precise communication protocol, including a specification of the set of states the communicating entities can be in and, for each state, the set of messages the entities can receive/send.**

#### **Case 1: TCP/IP**

The Internet provides two fundamental communication protocols: TCP for connection-oriented reliable communication and IP for best-effort connection-less communication. Sending an IP message is like sending a letter through the mail: the USPS will do its best to deliver it, without guarantees. The same holds for IP. TCP uses IP for message transport but augments it with the requirement that the receipt of every message must be acknowledged (using an IP ACK message) to the sender. An unacknowledged message will be resent.

Discussion Questions:

1. What states/messages are required for the IP protocol? What is the advantage of the IP protocol? What is the disadvantage?
2. What state/message types are needed, at a minimum, for the TCP protocol?
3. What state transitions can you describe?

4. What is the advantage of TCP? What is its cost?

### **Case 2: HTTP protocol**

Web browsers and web servers communicate using the HTTP protocol. The web server is waiting for client requests in a starting, listening state. A click of a hyperlink in a client-side browser will create a TCP connection between client and server. This connection will be used by the client browser to send a HTTP "GET" message to the server along with the url of the requested resource. The server will respond with a reply code and, if the request was successful, the requested content.

Discussion Question:

1. What happens when we click on a hyperlink? How does the browser connect to the web server hosting the resource? Should this be a reliable or unreliable connection?
2. What information should the request message contain?
3. What information should the reply message contain? How are errors handled?

**Assessment: Web browsers cache recently visited content. Future requests for the same content can be served from the cache. How long should the browser be doing this? Well, it depends on the content and the content owner (web server) determines this. How can you augment the HTTP protocol to let the web content server inform the web browser how long should the cached content be considered fresh?**