

# HARVARD CS50 AP

AP COMPUTER SCIENCE PRINCIPLES CONTENT REVIEW GUIDE



{ C }

*Programming*

**Johns Creek High School**  
**Cmdr. Raymond T. Schenk**  
**v1.0 11 December 2018**

This work is the student powered! My amazing students have come together to help assemble a work of information that is aligned by our course standards and put together to provide a fundamental platform for equipping future students at mastering the curriculum for this course.

Their hard work and leadership has made this entire guide possible, and they deserve the credit for amassing a wall of information, in service to their peers. I am forever grateful to their ability to leave a legacy for those come behind them.

HARVARD CS50 AP

P1 CONNECTED COMPUTING

P2 CREATING COMPUTATIONAL ARTIFACTS

CTAE Review: Professionalism

## Definition of *professionalism*

1: the conduct, aims, or qualities that characterize or mark a profession or a professional person

<https://www.merriam-webster.com/dictionary/professionalism>



### Things that show a lack of professionalism:

- Casual clothes, lack of effort in appearance
- Inappropriate language or volume of voice
- Missing deadlines with work
- Lack of motivation towards job responsibilities
- Blaming others for your mistakes
- Missing work
- Disrespectful behavior
- Unethical behavior or tolerance thereof

---

PROFESSIONALISM SHOULD BE MAINTAINED AT ALL TIMES IN THE WORKPLACE.

### Things that indicate strong professionalism:

- Strong work ethic
- Creativity
- Working towards the mission of the company or organization
- Cooperation
- Service to others
- Self-regulation
- Motivated productivity and ambition
- Strong ethical performance
- Humility
- Proper Planning
- Respect for others

---

### IMPACTS OF COMPUTING ON OUR ECONOMY, CULTURE, SOCIETY

The preparatory work and EXPLORE submission to the College Board cover this standard. Since our course is primarily a programming emphasis, the following is the only assessable content beyond what is done for the required submissions:

1. There can be all manner of innovations, but what makes an innovation a **computing innovation** is the input->processing->output cycle of actual information (specifically nameable data). If one cannot name the data involved in an innovation and explain how a computational device modifies or consumes said data - to produce some form of knowledge or information in the form of output, then the innovation is typically not a computing innovation. Rather, it is more likely an engineering or manufacturing innovation – even a scientific innovation. *All computing innovations can have their data*

*sets readily identified and explained throughout the data processing cycle of the innovation.*

2. One should never make extraordinary claims about an innovation. Few computing innovations will actually ever shift the GDP of the United States.
3. Using the colloquialism 'hacked' is decidedly improper when analyzing security risk. Trained computer science professionals specifically identify the risk, and the appropriate mitigations for that risk. For example, an exposed social security number is at risk for unauthorized access if it is not stored in an encrypted manner. The mitigation is to encrypt either the data itself at generation (best) or apply global encryption to the SSN field, table, or database.
4. When creating artifacts, it is critical to use attention to detail to provide the correct information, presented in the specific manner required. This skillset translates to every aspect of project management and even more importantly, to project proposals and bids.

### P3: ABSTRACTION

1. **User Interface/ User Experience (UI/UX)**- User interface includes all the parts of the website, apps, or technological device that interact with the user whereas user experience is the emotions and accessibility of a product. The user experience can be modified to be more efficient.
2. **Elements**- all the HTML components of a website
3. **Semantic Elements**- HTML elements which cannot be physically seen on a website
4. **Back End**- all the parts of a website that make it run including applications, web servers, and databases and is usually not visible to the user
5. **Bugs** - the coding mistakes or unnecessary pieces of code that prevent a website/program from working properly
6. **Frameworks** - collection of programs that developers can use to develop software (ex. Ruby on Rails and Bootstrap)
7. **Version Control** - tool used to store changes to code/files on a website or app which allows the user to restore previous versions in case of a bug
8. **Caching** - when a web browser stores information about a website (images/font/etc.) in order for it to load faster when a user revisits the page
9. **Cloud Computing** - the storing and processing of data among several remote servers accessible through the internet as opposed to locally on a computer
10. **Virtual Private Network (VPN)** - networks that allow private networks to use public internet connections in order to improve security
11. **Relational Database Management System (RDBMS)** - systems used to organize data into tables which can then be accessed without the need to reorganize the tables
12. **Hybrid app** - apps that work on different platforms and which are a combination of a native app (stored locally on a device) and a web app (accessible through the Internet)

---

## INFOSEC

**Computer Virus** – Malicious Software that can replicate itself

**Malware** – Any unwanted software on your computer

**Trojan Software** – Software that carries a malicious payload, while pretending to be normal software.

**Backdoor** - A backdoor Trojan is a loophole into the infected computer through remote control. The author gains complete access to, and control over all files and data. Backdoor Trojans are often used to unite a group of victim computers to form a botnet or zombie network that can be used for criminal purposes.

**Exploits**- Exploits are programs that contain data or code that takes advantage of a vulnerability within application software that's running on your computer.

**Rootkits**-Rootkits extend the longevity of viruses by concealing certain objects or activities in your computer system.

**Trojan-Banker** - designed to steal your account data for online banking systems, e-payment systems and credit or debit cards.

**Trojan-DDoS** - conduct DoS (Denial of Service) attacks against a targeted web address. By sending multiple requests – from your computer and several other infected computers – the attack can overwhelm the target address... leading to a denial of service.

**Trojan-Downloader** - download and install new versions of malicious programs onto your computer – including Trojans and adware.

**Trojan-Dropper** - Hackers use these programs to install Trojans, malicious programs and/or viruses undetected. Not all antivirus programs can scan all the components in these Trojans.

**Trojan-FakeAV** - They simulate activity of antivirus software. In reality they simply require money in return for reports of fake threats.

**Trojan-Ransom** - This type of Trojan can encrypt data on your computer – so that your computer doesn't run correctly, or you can no longer use specific data. The criminal will only restore your computer's performance or decrypt your data, after you have paid them the ransom money that they demand.

**Trojan-Spy** - Trojan-Spy programs can spy on how you're using your computer – for example, by tracking the data you enter via your keyboard, taking screenshots or getting a list of running applications.

**Keyloggers** – Malicious code or spyware that logs and sometimes transmits every keystroke you make while operating your computer. Can be used to determine accounts, usernames, and passwords.

**Peer-to-Peer Network** – All computers equally share resources. Every computer is a peer.

**Client-Server Network** – Server enables multiple clients to share its resources. Functions are centralized and hosted by a set of network "servers." Servers also provide security.

**LAN** – Local Area Network – Single office, home, school, etc.

**CAN** – Campus Area Network. Because colleges want a fancy network name too!



**PAN** – Personal Area Network – Individual network (i.e. phone, laptop, tablet, wireless earbuds)  
(No association with the flying boys)

**WAN** – Wide Area Network – Wide geographically, often leased telecommunications services

**MAN** – Metro Area Network – Metropolitan network, single city or town.

**Wireless Access Point (WAP)** – Network device that wirelessly connects multiple devices to a given network

**Network Standards** – Rules that establish interoperability across networks. (OSI Model is largest set of network standards)

**Network Security** – Software and appliances on networks and computers to prevent and limit unauthorized access to resources.

**Network Address** – An identifier for a node on a network. Should be unique. Receiving and Internet Protocol Address is how one is considered to be “connected to the Internet.”

**Network Diagnostics** – Software designed to help users and network administrators monitor and troubleshoot connections to the network or Internet.

---

## NETWORK TOPOLOGIES:

**There are two main distinctions:**

1. **Physical Network** – How hardware is connected
2. **Logical Network** - How data flows

**Bus** – 1 Cable, everything connected to that. Single point of failure

**Ring** – Unit connected in a ring (Common in fiber), single point of failure

**Star** – Every bus connected to a central hub, single point of failure

**Full Mesh** – Everyone connected to everyone, most failsafe, multiple paths, **most expensive**

*With  $n$  devices in full mesh, total number of dedicated links is calculated by  $(n*(n-1))/2$*

*So if 5 devices are fully meshed,  $5*4/2 = 10$  **connections** required.*

---

## COMMON NETWORK HARDWARE

**Switch** – Comes in *smart* switches and *dumb* switches. Smart switches have additional software and configuration/monitoring capabilities. Dumb switches are plug-and-play, often for home and small office use. Old names include switching Hub, Bridging Hub, MAC Bridge.

**“Switches build networks.”**

**Routers** – Routers connect networks to other networks.

**Network Interface Card (NIC)** – Hardware that connects a computer to a network. Today most often part of motherboards or main boards.

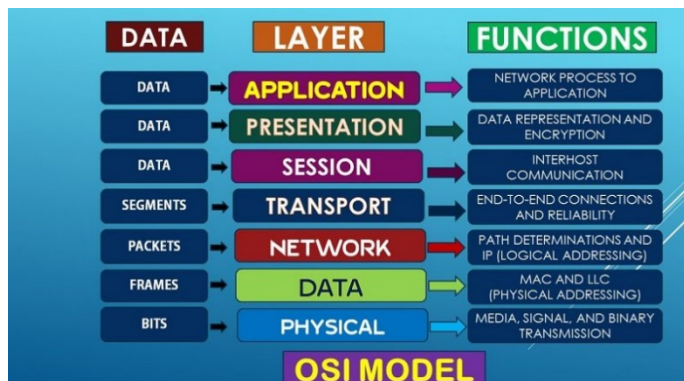
**Network Interface** – Software that connects a computing device to a network.

**Blacklisting/Whitelisting** – Firewalls either blacklist or whitelist. Whitelisting means no site traffic can access network unless it is on the whitelist. Blacklisting means any traffic can enter unless it is on the blacklist. Can also be applied to specific file formats for high-speed guards that do deep packet inspection.

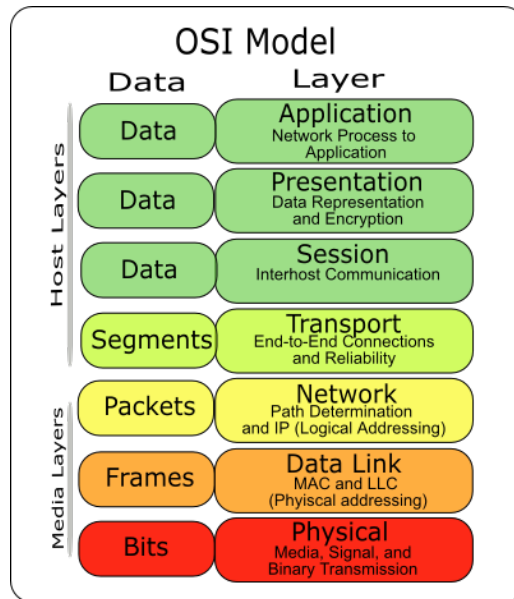
**Network Gateway** – allows on LAN to link to another LAN

---

## THE OSI MODEL FOR THE INTERNET



<https://proxy.duckduckgo.com/iu/?u=https%3A%2F%2Fkailashafoundation.org%2Fwp-content%2Fuploads%2F2017%2F06%2Fosi-2-806x440.jpg&f=1>



<https://proxy.duckduckgo.com/iu/?u=https%3A%2F%2Fnetworksmania.files.wordpress.com%2F2012%2F03%2Fdata-link.png&f=1>

### Layer 7: Application Layer

Services end-user applications. Works with the data, as in a web browser, to display or send information via the presentation layer.

### Layer 6: Presentation Layer

Format conversions and encryption/decryption. This is the “file format” layer, where files are reconstituted or broken down depending on which direction of the stack information is moving.

### Layer 5: Session Layer

Manages sequence and flow that initiates and tears down connections across networks. Multiple types of connections can occur at Layer 5.

### Layer 4: Transport Layer (TCP)

This layer delivers data across the network connections. TCP is the most common example of a transport Layer 4 network protocol. TCP controls error recovery, flow control, and re-transmission controls. Also the Layer of Uniform Datagram Protocol (UDP) Packets.

### Layer 3: Network Layer (IP)

The network layer adds the concept of routing. Frame data is inspected to see if it has reached its destination. If so, the data is formatted up to the transport layer. IP Addresses are maintained at this layer. Mapping is achieved through the Address Resolution Protocol (ARP).



## Layer 2: Data Link Layer

The data link layer deals in frames. It is the most complex layer of the OSI model, it is often subdivided into two layers, the MAC layer and the Logical Link Control Layer (LLC)

## Layer 1: Physical Layer

The physical layer is responsible for the actual transmission of bits. It consists of the actual hardware required to achieve this, including cables, fiber, hubs, repeaters, microwave transceivers, etc.

### MORE TERMINOLOGY

---

**UDP Packets** – send streaming without the controls of TCP. Live broadcasts. Need fast arrival, but not the delays inherent in TCP guaranteed delivery.

**Packet** – formatted unit of data carried across packet-switching networks.

**Dynamic Host Configuration Protocol (DHCP) Server** – A server that assigns IP addresses either dynamically, or statically to devices wishing to join a network.

**Domain Name Service** – translates domain names into IP addresses so humans don't have to do so. This is how [www.mysite.com](http://www.mysite.com) becomes 60.123.22.123.

**IPv4** – Internet Protocol Version 4. Uses 32 bits. World running out of IP addresses despite using internal IP for individual networks. (4.3 billion addresses) Format: 255.255.255.255

**IPv6** - Internet Protocol Version 6. Uses 128 bits. 8 Words (8 x 2-bytes) separated by colons. (3.4x10<sup>38</sup> available addresses). Format: FF:FF:FF:FF:FF:FF:FF:FF

**Secured Socket Layer** – hackable precursor to TLS security.

**Transport Layer Security (TLS)** – Replaced SSL, is a stateful connection. Provides:

- Data encrypted symmetric encryption/decryption
- Authentication via the encryption
- Reliable Data-Message Authentication Codes (MAC ID) to detect unauthorized data modification or loss

---

## THE FIRST ABSTRACTION: HOW COMPUTERS REPRESENT DATA

**ANSI** - American National Standards Institute – American non-profit organization that develops and maintains standards for industry.

**ASCII** – American Standard for Information Interchange. Written and developed/maintained by ANSI. ASCII was the world's answer to defeat IBM's EBCDIC

### ASCII DIGITS

---

011 0001	1
011 0010	2
011 0011	3
011 0100	4
011 0101	5

### ASCII CHARACTERS (UPPER CASE)

---

100 0001	A
100 0010	B
100 0011	C
100 0100	D
100 0101	E

### ASCII CHARACTERS (LOWER CASE)

---

110 0001	a
110 0010	b
110 0011	c
110 0100	d
110 0101	e

### ASCII CHARACTERS (SPECIAL CASE)

---

000 0000	Null	(Blank tape)
111 1111	Delete	(All holes punched)

- Computers use 7 bits for ASCII (8 bit is 0) – Fits in a single byte
- Extended ASCII uses the extra bit to add more characters but contains ASCII perfectly.
- UNICODE uses between 1 and 4 bytes, *usually* containing a character. It contains all of ASCII and a lot more characters and combinations of accents. It **does not** contain every character of every language, but many say it tries to do so.

**Storage Area Network (SAN)** – an array of hard drives that ship only with low level block I/O capabilities. Must be formatted and initialized under a specific operating system's disk management protocols. These units are typically very large and expensive.

**Network Accessible Storage (NAS)** – more affordable hard drive space that can be accessed by users on a network. Many home and small offices use RAID-based NAS devices for storing and/or backing up valuable files.

**Virtual Private Network** – A network that allows users to access its resources via an encrypted session. VPNs provide security to an organizations network but can be considered a vulnerability to networks hosting users who use them.

#### P4: ANALYZING PROBLEMS AND ARTIFACTS

##### PSEUDO CODING

---



###### Advantages of pseudocode:

- Pseudocode is understood by the programmers of all types.
- It enables the programmer to concentrate only on the algorithm part of the code development.
- It cannot be compiled into an executable program.

**Pseudocode** is basically just writing down the logic of your solution to a specific coding using plain English. Pseudocode is a useful thing to learn because it focuses on the building block concepts of programming languages without you having to worry about your word choice

##### FLOW CHARTING

---

Symbol	Purpose	Description
	Flow line	Used to indicate the flow of logic by connecting symbols.
	Terminal(Stop/Start)	Used to represent start and end of flowchart.

	Input/Output	Used for input and output operation.
	Processing	Used for arithmetic operations and data-manipulations.
	Decision	Used to represent the operation in which there are two alternatives, true and false.
	On-page Connector	Used to join different flowline
	Off-page Connector	Used to connect flowchart portion on different page.
	Predefined Process/Function	Used to represent a group of statements performing one processing task.

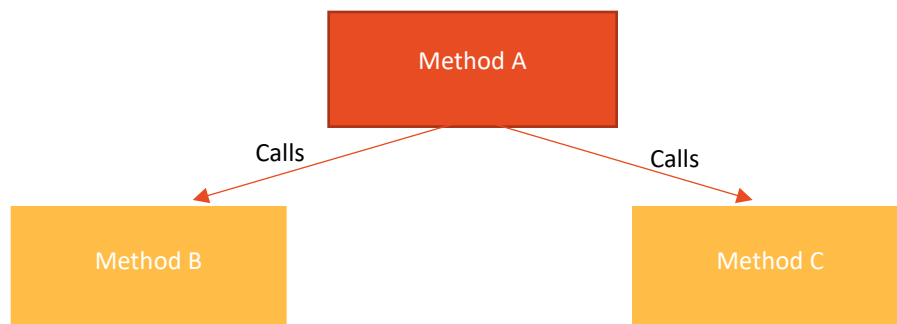
---

## ABSTRACTIONS IN USE

Any method is an abstraction. A method is a named series of instructions that can be called repeatedly. It can have a return value (function) or simply return nothing - void (procedure). Either way, the named method can also accept input parameters. In the definition of the method, these parameters are called **formal parameters**, because they are merely variable names scoped to the internal domain of the method. The **actual parameters** are the values passed into the method via the formal parameters whenever the function is called.

If a method calls another method, this is tiering methods, or a “higher order” of abstraction. When we import libraries of classes and other methods from other programmers to use, we are in essence standing on the shoulders of those who have gone before us. These layers of abstractions are what make programming as powerful as it is today.

For CREATE, the college board will require a small pyramid tier of abstractions, which can be visualized in the following diagram:



**IN THIS DIAGRAM, A SINGLE METHOD CALLS TWO OTHER METHODS. OF THE TWO OTHER METHODS, THE EACH SHOULD CONTAIN CALCULATIONS AND FLOW-CONTROL LOGIC. ADDITIONALLY, ONE OF THEM (B OR C) MUST BE USED REPEATEDLY IN THE PROGRAM TO “HELP REDUCE COMPLEXITY OF THE PROGRAM.”**

For CREATE, an abstraction must contain at least one calculation and some form of control flow logic. Students will practice setting up this relationship prior to CREATE.

P5: COMMUNICATION

P6: COLLABORATING

Students will perform all manner of design thinking and AGILE development practice, which will more than suffice for depth coverage of these two principles. Agile development SDLC will be covered after the C language concepts are presented below.

## C LANGUAGE NOTES FOR ASSESSMENT

C is an extremely small language, written by Dennis Ritchie between 1969 and 1973 at Bell Labs. It was used to implement the UNIX operating system.

C is the **most widely used language of all time**.

C is statically typed. C supports structured programming, lexical variable scope, and recursion.

**Lexical Variable Scope** – Variable naming is always referencing the local scope of a variable. Also known as statically scoped. This makes variables easier to reason about. This term opposes dynamic scoping, in which programmers must anticipate all possible dynamic contexts in which a module's code might be invoked.

**Recursion** – When a procedure calls itself. Requires a base case that returns a simple value to unwind the recursion. All recursive functions can be solved iteratively.

Pascal and ALGOL were other early languages to also use static scoping.

C's compiler is called Clang (C Language)

C can be used to define types (typedef) and precursors to classes called struct.

Supports ENUMs as well as null-terminated arrays of characters (strings)

Argc is the array length of the char \* known as argv.

Argv is the array of "strings" that are the command-line parameters.

**For example, the command line**

```
$ ./hello Ray Blue
```

**Would have an argc of 3 (3 elements), and an argv of:**

```
./hello  argv[0]
```

```
Ray      argv[1]
```

```
Blue     argv[2]
```



---

## SEQUENCING

Sequencing in C is creating steps to solve a problem in the correct order. This is exemplified in the PB&J demo during the opening event for this class. Order matters.

---

## SELECTION

Selection is picking an execution path through code. Determined typically by a boolean or multiple outcome control statement. Examples include If...Else. Or a switch statement.

---

## ITERATION

There are three types of loops: for loops, do-while and while-do loops.

For loops are for knowable iterations, providing in the syntax, an iterator with an initial point, a stopping point, and how much to skip each time to advance or retreat the pointer.

Do while loops are for when it is simply not known how long a loop must continue, but for which the first execution of the loop is mandatory, as in a non-graphical text-based menu, where the menu must be seen before anything is decided.

While-Do loops are the inversion of Do-Whiles. They simply check for the boolean true before ever executing a single line of iteration code. This means that it is possible never to get inside a while-do loop.

Switch statements allow for multiple paths to be available based on a value with more outcomes than a simple boolean. Switches are quite powerful and can simplify complicated decision code dramatically. They should be used carefully with break statements to help the code fall through correctly without executing too much of the code under the switch.

---

## ENCRYPTION/DECRYPTION

Encryption requires *plaintext*, a *cipher*, and a *key*.

Decryption requires *ciphertext*, the *cipher used for encryption*, and a *key*.

Keys can be *symmetric*, or *asymmetric* (public private)

## GENERAL TERMS

---

**Call** - location in the code where a method is called.

**Invoke** - to activate or begin a function in a line of code

**Variables** - memory location with a name token used to store a specific data type.

**Reference Variables** - data type that points to an object in memory

**Data Type** - the classification of what data is being stored. Includes primitives or custom (objects).

**Return Value** - the value a method returns.

**Programming Language** – a formal language made up of a grammar, syntax, and a set of instructions used to produce various kinds of output.

**Software Reuse** - the process of creating software systems from existing software. (Leveraging previous abstractions)

**Pascal Notation** - naming in which the first letter in each word in a compound name is capitalized. Class and Interface names use Pascal notation.

**Camel Case** - naming in which the first letter in each word in a compound is capitalized ***except for the first letter in the name.*** Variable and method names use Camel Case, aka Camel-hump naming.

**Big O Notation** – Big O() is the **worst**-case analysis of an algorithm, typically used in computer science to compare sorting algorithms or major traversals of ADS (Abstract Data Structures).

- Bubble, Insertion, Selection and Quick sorts are all  $O(n^2)$
- Merge is  $O(n\log(n))$  (Faster for larger data sets because of recursion)

**Omega Notation** – Omega notation is **best**-case performance, in the same light as O().

- Bubble, Insertion  $\Omega(n)$
- Selection  $\Omega(n^2)$
- Merge, Quick,  $\Omega(n\log(n))$

**Harmful Effect** - a change that occurs as a result of something else that has negative results.

**Beneficial Effect** - a change that occurs as a result of something else that has positive results.

**Intended Purpose** - the use for which something is designed.

## EXTENDED LEARNING

### Software Development Life-Cycle (SDLC)

**Waterfall** – oldest, slowest, very inflexible, Linear

**Others:** Prototyping, Spiral

**Best:** Agile – cyclical and iterative

**Individuals and Interactions** over processes and tools

**Working Software** over comprehensive documentation

**Customer Collaboration** over contract negotiation

**Responding to Change** over following a plan

**Runs in Sprints, is Iterative, Incremental and evolutionary**

**Scrum** is a specific software development agile methodology. 3-9 members. 2-4 weeks max sprints. Daily scrums are 15-min re-planning meetings.