



## FIRST TERM Phase 1

First term should be a microcosm of the entire 2 years of Computer Science.

Core Concepts and CS ideas should, at all opportunities, be experienced.

The **ABCDE** of Core Concepts needs to be emphasised, in proportion to their relevance to the topic.

(**A**bstraction, **B**asic algorithms, **C**omputer systems, **D**ata, **E**valuation and testing)



## RESOURCEFULNESS

Student needs to be encouraged into being resourceful.  
Why?

- It is how programmers and computer scientists work.
- A key skill of the LC and a principal objective.

The **ABCDE** of Core Concepts needs to be emphasised, in proportion to their relevance to the topic.

## Approaches to address LOs across all 3 strands

LCCS 2018

1. Learn how to transition from solving a problem, using say intuition or logic or maths, to solving the problem in a computational manner, that can be readily translated into an algorithm.
2. Learn key programming concepts in Python and also in some block coding platform such as Scratch.
3. Develop microprocessor skills to scaffold students for future Applied Learning Tasks.
4. Work individually on tasks to deepen understanding of how to design, debug and test algorithms.
5. Work collaboratively on tasks, both plugged and unplugged, to improve teamwork, design and CT skills.

# Cross Cutting Methods

Students will spend the first term learning how to think computationally, and in the process learn about:

**A. Abstraction... How did you transform the initial information into a generalised solution?**

B. Basic Algorithms... Create a set of instructions to solve a problem in a manner that computers can solve.

C. Computer Systems... how does it all come down to ones and zeroes?

D. Data... handling data, analysing data, storing data, ...

E. Evaluation.. Testing your programs and checking the logic. How do you know your program is working to solve the initial problem, in all cases?

# Cross Cutting Methods

Students will spend the first term learning how to think computationally, and in the process learn about:

- A. Abstraction... How did you transform the initial information into a generalised solution?
- B. Basic Algorithms... Create a set of instructions to solve a problem in a manner that computers can solve.**
- C. Computer Systems... how does it all come down to ones and zeroes?
- D. Data... handling data, analysing data, storing data, ...
- E. Evaluation.. Testing your programs and checking the logic. How do you know your program is working to solve the initial problem, in all cases?

# Cross Cutting Methods

Students will spend the first term learning how to think computationally, and in the process learn about:

- A. Abstraction.... How did you transform the initial information into a generalised solution?
- B. Basic Algorithms... Create a set of instructions to solve a problem in a manner that computers can solve.
- C. Computer Systems... how does it all come down to ones and zeroes?**
- D. Data... handling data, analysing data, storing data, ...
- E. Evaluation.. Testing your programs and checking the logic. How do you know your program is working to solve the initial problem, in all cases?

# Cross Cutting Methods

Students will spend the first term learning how to think computationally, and in the process learn about:

- A. Abstraction.... How did you transform the initial information into a generalised solution?
- B. Basic Algorithms... Create a set of instructions to solve a problem in a manner that computers can solve.
- C. Computer Systems... how does it all come down to ones and zeroes?
- D. Data... handling data, analysing data, storing data, ...**
- E. Evaluation.. Testing your programs and checking the logic. How do you know your program is working to solve the initial problem, in all cases?

# Cross Cutting Methods

Students will spend the first term learning how to think computationally, and in the process learn about:

- A. Abstraction.... How did you transform the initial information into a generalised solution?
- B. Basic Algorithms... Create a set of instructions to solve a problem in a manner that computers can solve.
- C. Computer Systems... how does it all come down to ones and zeroes?
- D. Data... handling data, analysing data, storing data, ...
- E. Evaluation.. Testing your programs and checking the logic. How do you know your program is working to solve the initial problem, in all cases?**