



Introduction to Computer Science High School

1.0 Principles of Computer Organization.

- 1.1 Identify the various functional components of a computer.
- 1.2 Match a list of computer terms and definitions/functions.
- 1.3 Describe the interaction of the various functional components of the computer.
- 1.4 Make appropriate decisions when purchasing a computer for home use.

2.0 Problem Solving.

- 2.1 Name and explain the steps in the problem-solving process.
- 2.2 Solve a problem by applying the problem-solving process.
- 2.3 Express a solution using standard design tools.
- 2.4 Determine if a given algorithm successfully solves a stated problem.

3.0 Connections between Mathematics and Computer Science.

- 3.1 Convert between decimal, binary, and hexadecimal numbers.
- 3.2 Create and translate an encoded message using a simple method of encryption.
- 3.3 Write conditional statements that include simple and complex Boolean expressions to solve stated problems.
- 3.4 Convert a problem description into correct set notation and apply appropriate set operations.
- 3.5 Write a function (method, etc.) that returns the correct value, given a function definition in mathematical notation.

4.0 Ethical Issues.

- 4.1 Distinguish between ethical and legal issues in a case study by listing the issues that can be resolved through the legal system and those issues that cannot be legally resolved.
- 4.2 Defend an ethical stance given a controversial or ethically ambiguous situation in a debate.
- 4.3 List and explain at least two positive and negative effects of one technological innovation on human culture.
- 4.4 Define intellectual property and state the impact of provisions to protect it.
- 4.5 Identify at least two benefits and two drawbacks of using commercial, public

domain, open source, and shareware.

4.6 Demonstrate behavior in the use of technology that conforms to school and local code.

5.0 Programming Languages.

5.1 Code, test, and execute a program that corresponds to a set of specifications.

5.2 Convert a word problem into code using top-down design.

5.3 Select appropriate data types.

5.4 Write structured program code.

5.5 Draw a series of diagrams showing the scope and values of variables during execution of a simple program.