Indiana CTE

Computer Science I

Problem Analysis

1 Students analyze a problem and develop a solution by creating a computer program.

1. Identify how to use a computer program to solve a problem `CS1-1.1`
2. Construct interactive computer programs that accept various forms of input and produce various forms of output, as a solution to a computer programming problem `CS1-1.2`
3. Use print charts, file layouts, program narratives, hierarchy charts, and system flowcharts, which accurately depict the problem assigned and describe the solution `CS1-1.3`
4. Report the program schematics and usage `CS1-1.4`
5. Identify the standard program flowchart symbols and use them correctly within the context of the basic control structures of sequence, selection and looping `CS1-1.5`

Software Tools

1 Students apply and adapt software tools to develop a computer program.

1. Construct a program that processes information `CS1-2.1`
2. Identify programming languages as procedural or object oriented `CS1-2.2`
3. Develop programs using reusable modules (modularization) `CS1-2.3`
4. Use debugging techniques to correct and validate the computer program `CS1-2.4`
5. Construct the program in a high-level programming language based on a created design `CS1-2.5`
6. Construct a program that opens and closes a file `CS1-2.6`

Algorithm

1 Students design a solution to the problem using algorithms.

1. Develop algorithms to solve a computer programming problem(s) `CS1-3.1`
2. Assess the use of algorithms to provide a solution to a programming problem `CS1-3.2`
3. Use pseudo code to describe a solution to a programming problem `CS1-3.3`
4. Create a program flowchart and ANSI standard flowcharting symbols to define a solution to a programming problem `CS1-3.4`
5. Explain how the algorithm can be used to solve a problem `CS1-3.5`

Program Development

1 Students create a functional computer program.

1. Define the process of programming `CS1-4.1`
2. Create a computer program that corresponds to an algorithm or proposed solution `CS1-4.2`
3. Define programming structures `CS1-4.3`
4. Recognize data variables and constants `CS1-4.4`
5. Recognize local scope and global scope `CS1-4.5`
6. Use conditionals (IF statements) `CS1-4.6`
7. Use loops (while statements, for statements) `CS1-4.7`
8. Define single and multidimensional Arrays `CS1-4.8`
9. Use functions and methods to break down the program logic and support reuse `CS1-4.9`
10. Define the graphical user interface `CS1-4.10`
11. Identify the parts of the programming platform `CS1-4.11`
12. Identify different types of errors and handle them programmatically `CS1-4.12`
13. Use the order of operations when using calculations `CS1-4.13`

Program Verification and Debugging

1 Students prove a computer program solution works by using verification and debugging techniques.

1. Predict and explain output `CS1-5.1`
2. Identify cause/effect for input/output `CS1-5.2`
3. Perform input validation `CS1-5.3`
4. Scrutinize peers code for errors `CS1-5.4`

---

6 Students connect an associated task with the code by providing documentation.

1. Describe the function of a computer program
2. Identify the purposes of a computer program
3. Explain concepts related to a computer program
4. Describe how to use a computer program
5. Identify cause/effect by explaining input and output
6. Interpret input/output