Mason City Schools

**Business Technology Department**

**Gregory D. Kummer**

Computer Programming I

Topic: Computer Science and Information Systems

Outcome: Students will understand beginning programming concepts related to the science of computer operations. They will demonstrate competency by applying these concepts to the development of computer programs.

Grade Level: 9-12

Length of Course: one trimester

Prerequisite: Ability to correctly type 25 words per minute. Must have completed Algebra I with a grade of C+of better, or consent of instructor.

| **SUBCONCEPTS**  (What will students know?) | **INTERACTIVE SKILLS**  (What will students be able to do?) | **RESOURCES** |
| --- | --- | --- |
| Programming languages can be used to output information from a computer and to input data into a computer. |  Output a line of text to the screen. | *Visual BASIC 6.0 Professional for Windows*. |
|  |  Define real, integer, and string data types. |  |
|  |  Use real, integer, and string data types in programming problems. |  |
|  |  Input data to a program from the program and from the keyboard. | *Using Visual BASIC* by M. Sprague. Southwestern. |
| Algorithms can be designed to create solutions that are correct, robust, reliable, and efficient. |  Write algorithms for classical mathematical problems. | Teacher handouts, sample code, and rubrics for each project in StudentWork folder- Computer Programming I folder. |
|  |  Write algorithms for classical computer science problems. |  |
| Programs should be well-designed using a problem-solving, top-down design process that minimizes corrections. |  Define the scope of a programming problem using a top-down design, with its needed inputs and outputs, before coding the problem. |  |
|  |  Refine the problem and algorithm. |  |
|  |  Code the program. |  |
|  |  Debug the program. |  |
| Programs should be well-designed using a problem-solving, top-down design process that minimizes corrections (continued) |  Validate the program, testing at breakpoints. |  |
| Programming languages can be used to create conditional statements. |  Define the relational operators. |  |
|  |  Define the logical operators. |  |
|  |  Use relational and logical operators to create simple and complex conditional statements. |  |
| Programming languages can be used to create control structures. |  Use counters and accumulators. |  |
|  |  Use sequencing control structures. |  |
|  |  Use selection control structures such as if/else and CASE |  |
|  |  Use repetition control structure including iterative structures, while structures, and until structures. |  |
| Programming languages can be used to manipulate text. |  Output text to various screen locations. |  |
|  |  Eliminate specified characters from a text string. |  |
|  |  Concatenate text strings. |  |
|  |  Compare text strings. |  |
| Programming languages can be used to create graphics and sound. |  Create alpha-numeric based graphical output. |  |
|  |  Identify simple graphic commands. |  |
|  |  Use sound in a program. |  |
| Programming languages can be used to create modules that are reusable. |  Use procedures to implement portions of a programming problem. |  |
|  |  Use functions to implement limited algorithms. |  |
| Program interfaces (Human-Computer Interfaces) can be well-designed and easy to use. |  Write programs in a friendly style. |  |
|  |  Trap for input common errors. |  |
|  |  Trap for unsuitable or unanticipated inputs. |  |
| Programs should be written in a well-formatted style. |  Document variables. |  |
|  |  Demonstrate a modular programming style. |  |
|  |  Employ self-explanatory output. |  |
| Operating systems are used for file management. |  Use an operating system to:   copy files.   delete files.   move files.   recover files.   create subdirectories.   remove subdirectories. |  |
| Computer Science is an evolving field. |  Review periodical literature to keep current and adapt to change. | Periodical subscriptions and/or Internet access |
|  |  Describe how specification changes and technological advances require the modification of programs. |  |
| Computer Science has historical and cultural issues. |  Trace the historical development and social implications of computer science. |  |
|  |  Draw conclusions regarding the issues of access, privacy, and ethics. |  |
| Artificial intelligence and robotics continues to impact society. |  Define artificial intelligence and robotics. | Video series |
|  |  Identify common uses of artificial intelligence and robotics in society. |  |
|  |  Explain the impact of artificial intelligence and robotics on society. |  |