Mason City Schools

**Mathematics Department**

Gregory D. Kummer

*HONORS COMPUTER PROGRAMMING II*

Strand: Computer Science and Information Systems

Outcome: Students will understand advanced programming concepts related to the science of computer operations using Visual Basic .Net 2010. They will demonstrate competency by applying these concepts to the development of computer programs.

Grade Level: 9-12

Length of Course: one trimester

Prerequisite: “B” or better in Computer Programming I and consent of instructor

| **SUBCONCEPTS/INDICATOR**  (What will students know?) | **PERFORMANCE OBJECTIVES**  (What will students be able to do?) | **METHODS & MATERIALS** |
| --- | --- | --- |
| Programming languages can be used to output information from a computer and to input data into a computer. |  Define Boolean and abstract data types. | Bradley, Julie Case and Anita C. Millspaugh**. *PROGRAMMING IN VISUAL BASIC 2010,* Boston, MA: McGraw-Hill, Copyright 2011, ISBN-13: 978-0-07-351725-4. Book Web Site** [**http://www.mhhe.com/VB2010/**](http://www.mhhe.com/VB2010/) |
|  |  Use Boolean and abstract data types in programming problems. | Robin A. Reynolds-Haertle,  **OOP with Microsoft Visual Basic .NET and Microsoft Visual C# Step by Step,** |
|  |  Input data to a program from binary and ASCII data files. | Microsoft Press © 2002 , ISBN: 0735615683 |
| Algorithms can be designed to create solutions that are correct, robust, reliable, and efficient. |  Write algorithms for classical mathematical problems. |  |
|  |  Write algorithms for classical computer science problems. |  |
| Programs should be well-designed using a problem-solving, top-down design process that minimizes corrections. |  Independently define the scope of a programming problem using a top-down design, with its needed inputs and outputs before coding the problem. |  |
|  |  Independently refine the problem and algorithm. |  |
| Programs should be well designed using a problem-solving, top-down design process that minimizes corrections. (Continued) |  Use pseudocode to communicate programming ideas to others. |  |
|  |  Independently code the program. |  |
|  |  Independently debug the program. |  |
|  |  Independently validate the program, testing at breakpoints. |  |
|  |  Evaluate case-study programs. |  |
|  |  Organize personal program design into a case-study model. |  |
| Programming languages can be used to create routines to sort and search data. |  Define sorting algorithms and their efficiency. |  |
|  |  Define searching algorithms and their efficiency. |  |
|  |  Use sorting algorithms to arrange data in ascending and descending order. |  |
|  |  Use searching algorithms to find data in a sorted and unsorted list. |  |
|  |  Use searching algorithms to insert and delete data from a sorted list. |  |
| Programming languages can be used to create single and multiple dimension arrays. |  Create and enter data into a single-dimensional array. |  |
| 11-4-N – Use matrices to represent given information in a problem situation |  Manipulate data stored in a single-dimensional array. |  |
| 11-6-N – Compute sums, differences, and products of matrices |  Create and enter data into a two-dimensional array. |  |
| 11-7-P – Model and solve problems with matrices and vectors |  Manipulate data stored in a two-dimensional array. |  |
| Programming languages can be used to create single and multiple dimension arrays. (Continued) |  Understand the structure and application of arrays beyond two dimensions. |  |
| Programming languages can be used to solve problems with recursive algorithms. |  Define recursion. |  |
| 11-1-P – Identify and describe problem situations involving an iterative process that can be represented as a recursive function |  Identify recursion in examples. |  |
|  |  Write a recursive subroutine with teacher help. |  |
|  |  Identify problems which are best solved using recursive algorithms. |  |
| Programming languages can be used to create and manipulate records and sets. |  Define records and sets. |  |
|  |  Identify uses of records and sets in programming situations. |  |
|  |  Use records and sets to solve a programming problem. |  |
| Programming languages can be used to create graphics and sound. |  Apply draw commands to a variety of programs. |  |
|  |  Create an inventory of graphical images. |  |
|  |  Experiment with a variety of graphic commands. |  |
|  |  Demonstrate appropriate use of sound in computer programs. |  |
|  |  Synthesize a song from teacher provided list and incorporate it into a computer program. |  |
| Programming languages can be used to create modules that are reusable. |  Independently and consistently use procedures to implement programming problems. |  |
|  |  Independently and consistently use functions to implement algorithms. |  |
|  |  Modularize all programming to the greatest possible extent. |  |
| Programming languages can be used to create modules that are reusable. (Continued) |  Reuse modularized portions of code. |  |
|  |  Define the scope of a variable. |  |
|  |  Define the hierarchy of a modularized program. |  |
|  |  Create a group program using object-oriented design methods. |  |
| Program interfaces (human-computer interfaces) can be well designed and easy to use. |  Independently write programs in a friendly style, trap common input errors, unsuitable input, or unanticipated input. |  |
|  |  Implement the use of menus and mouse operations in a driver program. |  |
| Programs should be written in a well-formatted style. |  Document variables and remark code lines. |  |
|  |  Create a header program to automatically document programmer name, date, and class. |  |
|  |  Organize output into columns and tables whenever appropriate. |  |
| Programs used for file management. |  Use a program to:   create and manipulate text (.txt) files.   create and manipulate sequential (.dat) files.   create and manipulate record (.dat) files. |  |
| Computer Science is an evolving field. |  Review periodical literature to keep current and adapt to changes in Computer Science. | Periodical subscriptions and/or Internet access |
|  |  Hypothesize specification changes and technological advances which would require the modification of current programs. |  |
| Computer Science has historical and cultural issues. |  Identify the generations of computers and the people historically associated with each generation. |  |
|  |  Predict areas where computers will have a greater impact on the social fabric of the world. |  |
|  |  Communicate to others the issues of access, privacy, and ethics in regard to computers and their applications. |  |
| Artificial intelligence and robotics continue to impact society. |  Identify unusual uses of artificial intelligence and robotics in society. |  |
|  |  Predict future uses of artificial intelligence and robotics in society. |  |
|  |  Explain the impact of artificial intelligence and robotics on society. |  |