I. **OVERVIEW**
   The following information will appear in the 2010 - 2011 catalog

   **CMPSC 213 Programming With Visual Basic**
   **3 Units**

   **Prerequisite:** Satisfactory completion of CMPSC 204 with a minimum grade of C or better.

   Concepts in programming a computer using the language called Visual BASIC. Emphasis on structured design, graphical user interfacing, and documentation. Includes user screen development, control constructs, array processing, elementary file processing, and database access. Hands-on experience using microcomputers. Extensive interaction with computers will be expected.

   Three maximum completions.
   Field trips might be required. *(A-F or P/NP - Student choice) Lecture /Lab*
   **Transfer:** (CSU, UC) **General Education:** (MJC-GE: D2)

II. **LEARNING CONTEXT**
   Given the following learning context, the student who satisfactorily completes this course should be able to achieve the goals specified in Section III, Desired Learning:

   **A. COURSE CONTENT**

   1. **Required Content:**

      A. Windows Operating System (current version)
      1. Cursory overview of mouse operation.
      2. File selection
      3. Icon invocation
      B. Visual Basic Software
      1. Installation
      2. Configuration
      3. IDE (Integrated Development Environment)
      4. Project Structure
      C. Designing Applications
      1. Object oriented/event driven approach
      2. Building user interfaces
      3. Coding, testing and debugging applications
      4. Manipulating object properties
      D. Variables And Constants
      1. Variable types
      2. Variable and constant naming conventions
      3. Creating variables
      4. Variable Scope
      5. Option explicit
      6. Symbolic constants
      7. Manipulating values
      8. Variable array
      E. Visual Basic Language Constructs
      1. Sequence
      2. Selection
      3. Repetition
      4. Event handling
      5. Object structure
      6. Classes
      7. Code module
8. Class module
9. Error trapping

F. The Graphic User Interface (GUI)
1. Forms
2. Dialog Boxes
3. Input Boxes
4. Menus
5. Windows Common Controls
6. Buttons
7. Text
8. Graphics
9. Counters
10. ActiveX Controls
11. Control Array

G. Intrinsic Functions
1. Val
2. InputBox
3. Instr
4. Left, Right, Mid
5. Ucase, Lcase
6. Random

H. Files
1. Creation (sequential, random access)
2. Updating
3. Sorts
4. Merges

I. Database
1. Design
2. Structure
3. Accessing and updating
4. ADO control

J. Program Documentation
1. Remarks
2. TOE Charts
3. Structure charts
4. Data flow diagrams
5. Pseudocode
6. GUI layout sketches

2. **Required Lab Content:**

A. The Graphic User Interface (GUI)
1. Forms
2. Dialog Boxes
3. Input Boxes
4. Menus
5. Windows Common Controls
6. Buttons
7. Text
8. Graphics
9. Counters
10. ActiveX Controls
11. Control Array

B. Intrinsic Functions
1. Val
2. InputBox
3. Instr
4. Left, Right, Mid
5. Ucase, Lcase
6. Random

C. Files
1. Creation (sequential, random access)
2. Updating
3. Sorts
4. Merges  
D. Database  
1. Design  
2. Structure  
3. Accessing and updating  
4. ADO control  
E. Program Documentation  
1. Remarks  
2. TOE Charts  
3. Structure charts  
4. Data flow diagrams  
5. Pseudocode  
6. GUI layout sketches  

B. ENROLLMENT RESTRICTIONS  

1. Prerequisites  
Satisfactory completion of CMPSC 204 with a minimum grade of C or better.  

2. Requisite Skills  
*Before entering the course, the student will be able to:*  

a. Discuss the uses of computers and their effect on society.  
b. Describe the functional hardware components of a computer system.  
c. Describe the concepts and functions of a computer operating system and application software.  
d. Demonstrate the use of the Windows operating system to format storage, create folders, launch applications, navigate through multiple windows, and access menus and toolbars to accomplish specified tasks.  
e. Perform simple problem analysis, and design solution algorithms.  
f. Demonstrate the implementation of sequence, selection, and iterative control processes using a high-level programming language such as Visual Basic or Java.  
g. Discuss and apply the principles of top-down problem decomposition.  

C. HOURS AND UNITS  

<table>
<thead>
<tr>
<th>INST METHOD</th>
<th>TERM HOURS</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lect</td>
<td>36</td>
<td>2.00</td>
</tr>
<tr>
<td>Lab</td>
<td>54</td>
<td>1.00</td>
</tr>
<tr>
<td>Disc</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

3 Units  

D. METHODS OF INSTRUCTION (TYPICAL)  
*Instructors of the course might conduct the course using the following method:*  

1. Formal lectures by a certified instructor  
2. Assigned reading and discussion of required text  
3. Assigned reading of supplemental reference materials  
4. Implementation of computer laboratory projects  
5. Student will be required to apply standard systems analysis problem-solving methodology to all
programming projects. This methodology is described by the five phases of the program development cycle as follows:
Phase 1 – Written problem definition and program specifications
Phase 2 – Formal solution algorithm design using flowcharts, structure charts, or other standard algorithm presentation techniques
Phase 3 – Program code using valid syntax and constructs
Phase 4 – Implementation and critical testing and analysis of program results
Phase 5 – Written documentation, both internal and external, for the solution

E. ASSIGNMENTS (TYPICAL)

1. EVIDENCE OF APPROPRIATE WORKLOAD FOR COURSE UNITS
   Time spent on coursework in addition to hours of instruction (lecture hours)
   Weekly assignments using a visual programming design environment

2. In class assignments required to be finished on the computer during class lab time

3. Reading assignments should be on a textbook chapter basis, approximately one chapter and one chapter quiz per week. (Done online or at home as homework).

III. EVIDENCE OF CRITICAL THINKING
   Assignments require the appropriate level of critical thinking

Sample weekly assignment:

Assignment Submission: Lesson 7

Instructions

In this project you will do the 7th Lesson in the book, on pages 127-142 of the text. When you have read all the material and done all the steps, screenshot the final program as it looks when you are finished (figure 7-10). Also, find the file called frmDogYears.vb (created during the tutorial) and upload that.

Next, go to page 146 and do the section called Critical Thinking on that page. Upload the screenshot of the working program and rename the form file to "frmAnimalYears.vb" and upload it so I can grade your project.

After you get these 4 files uploaded, 2 screenshots and 2 vb files, submit the whole project for grading. Don’t forget to use paint to crop the screenshots and to save them as compressed jpg files.

Don’t forget to review all the materials in the Lesson, such as the sample tests, this material will be important on the next test.

Sample question from the Final:

91. (10 points) Create a program that has a textbox, a richtextbox, a checkbox and 3 buttons.

The textbox is where the user types a number. That number indicates how big the figure will be that is printed in the richtextbox.

The first button clears the richtextbox.
The second button makes a square in the richtextbox of asterisks. You must use loops to generate the image.

The third button makes a triangle in the richtextbox of asterisks. You must use loops to generate the image.

The checkbox makes stripes appear on the image. Print your form1.vb code and give a few screenshots. See the screenshot below for an example:

- TEXTS AND OTHER READINGS (TYPICAL)

- DESIRED LEARNING
  
  A. COURSE GOAL
     
     *As a result of satisfactory completion of this course, the student should be prepared to:*
     
     analyze and evaluate event-driven programming, evaluate and implement the principles of graphic user interface design, create input and output processes implemented in a high-level programming language, and create custom applications using Visual Basic.

  B. STUDENT LEARNING GOALS
     
     *Mastery of the following learning goals will enable the student to achieve the overall course goal.*

     A. Required Learning Goals
        
        *Upon satisfactory completion of this course, the student will be able to:*

        a. Recall elementary language constructs and their syntax.

        b. State concepts of object oriented, event driven programming and design.

        c. Define programming primitives such as counters, accumulators, and various sorting algorithms.

        d. Assess problems and design solutions using appropriate programming resources.

        e. Describe forms of notation used in documenting programs and illustrating algorithms.

        f. Outline methods of data storage including sequential and direct disk-file access and database storage.

        g. Define operating-system commands and shared system objects.

        h. Design problem-solving algorithms using modular and object oriented methodology.

        i. Construct algorithms in correct Visual Basic syntax using appropriate standardized object naming notation.

     B. Lab Learning Goals
        
        *Upon satisfactory completion of the lab portion of this course, the student will be able to:*

        a. Develop TOE chart application plans, pseudocode and draw flowcharts of algorithms.

        b. Construct algorithms in correct Visual Basic syntax using appropriate standardized object naming notation.
c. Produce program listings and output reports on computer printing equipment.
d. Evaluate (debug) their own programs or the programs of others.
e. Develop distribution strategies and create installation and distribution files.
f. Create graphical user interface applications that adhere to the GUI development standards outlined by Microsoft.
g. Construct algorithms in correct Visual Basic syntax using appropriate standardized object naming notation.
h. demonstrate updated skills reflecting current industry standards as software tools, interface and functions evolve in new versions.
i. demonstrate updated skills reflecting current industry standards as software tools, interface and functions evolve in new versions.
j. SECOND COMPLETION:

THIRD COMPLETION:

FOURTH COMPLETION:

demonstrate updated skills reflecting current industry standards as software tools, interface and functions evolve in new versions.

• METHODS OF ASSESSMENT (TYPICAL)

A. FORMATIVE ASSESSMENT
   
   A. Written unit examinations
   B. Demonstration of programming skills in lab
   C. Written final exam.
   D. Demonstration of interface design skills by submission of written design documentation

B. SUMMATIVE ASSESSMENT

   A. Written unit examinations
   B. Demonstration of programming skills in lab
   C. Written final exam.
   D. Demonstration of interface design skills by submission of written design documentation