

# **“Anyone can program” – Computer Programming Teaching Experience to the First Year Students**

---



UNIVERSITATEA  
DIN  
CRAIOVA



**Ion Buligiu, Amelia Bădică**

*Department of Business Informatics and Statistics*  
**University of Craiova, Romania**

**August 23, 2016**

**16th Workshop “Software Engineering Education and Reverse Engineering”**

# Talk Outline



- ❑ Really anyone can program?
- ❑ First step – Students preparation for programming
- ❑ Computer Programming Course Overview & Objectives
- ❑ Lab Content Structure
- ❑ Teaching Experience Course and Labs
- ❑ Conclusions

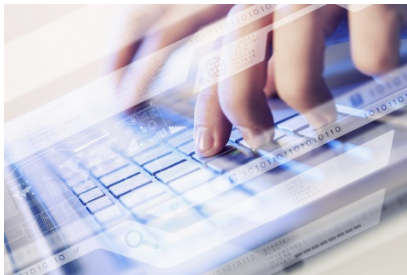
# Really anyone can program?

Yes, indeed – ANYONE can successfully write source code in a simple, intuitive and friendly language programming like Visual Basic, assisted by an advanced IDE interface provided by Visual Studio Community Edition (and others editions, of course ☺)



The secret is to begin with the simplest examples like “Hello, World!” and create basic visual interface with some objects placed into a form (like buttons, text boxes, labels and so on) and understand the hierarchy of objects, their properties, events responses and methods

Writing the code in Visual Studio is fully assisted by IDE through IntelliSense and AutoComplete and help the programmer to concentrate on logics in resolving the problem

A screenshot of the Visual Studio IDE interface. The top window shows the code for 'Form1.vb'. The code is in Visual Basic and defines a class 'frmHello' with two event handlers: 'btnHello\_Click' and 'btnClear\_Click'. The 'btnClear\_Click' handler is currently being edited, and the text 'txtMessage.clea' is visible. A dropdown menu is open, showing suggestions for 'Clear' and 'ClearUndo'. The 'Clear' option is selected, and a tooltip for 'Sub TextBoxBase.Clear()' is visible, stating 'Clears all text from the text box control.' The bottom window shows the 'Form1.vb [Design]\*' view, which is currently empty.

```
Form1.vb*  Form1.vb [Design]*
VB WindowsApplication20  btnClear  Click
Public Class frmHello
    Private Sub btnHello_Click(sender As Object, e As EventArgs) Handles btnHello.Click
        txtMessage.Text = "Hello, World"
    End Sub

    Private Sub btnClear_Click(sender As Object, e As EventArgs) Handles btnClear.Click
        txtMessage.clea
    End Sub
End Class
```

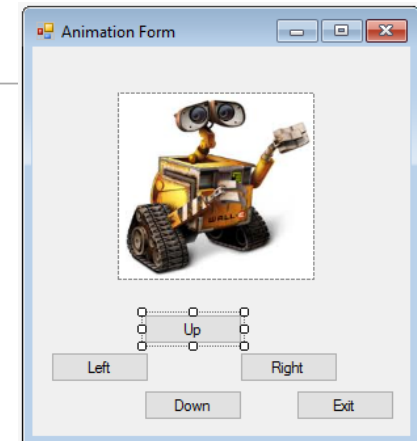
Clear  
ClearUndo

Sub TextBoxBase.Clear()  
Clears all text from the text box control.

## First step – Students preparation for programming

- ❑ Motivating students is the primary concern of the teacher to arouse their curiosity and open the appetite for study
- ❑ Ensure the certainty of writing source code using a simple and understandable language programming, such as Visual Basic, without having prior knowledge of programming
- ❑ Students must know the importance of knowing programming techniques and mastering a programming language in preparation for further study of other courses such as Object Oriented Programming, Data Structures, Computer Systems Design and others
- ❑ Through *Gamification* I tried to apply game elements and friendly examples in solving easy problems like animation of a picture into a window form by pressing four command buttons, one for each direction: up-down-left-right. The aim is to know the properties of the objects and which are the events that they respond and to determine the logical incremental code for each direction

```
Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Timer3.Tick
    If picPinguini.Height < (Me.Height - picPinguini.Top) - 20 Then
        picPinguini.Top = picPinguini.Top + 1
    Else
        Timer3.Enabled = False
    End If
End Sub
```



# Computer Programming Course Overview

- ✓ The Computer Programming Course is addressed to the 1<sup>st</sup> year students from Business Informatics, within Faculty of Economics and Business Administration
- ✓ It's no necessary prior knowledge about programming techniques or programming languages
- ✓ The course content and structure is designed according to the Computer Science curricula recommended by Association for Computing Machinery (ACM) and IEEE Computer Society
- ✓ The course prepares the students for the next year courses:
  - ❑ Object-Oriented Programming
  - ❑ Data Structures and Algorithms
  - ❑ Information Systems Design



# Computer Programming Course Objectives I

- ✓ The main purpose is to familiarize the students with the methods and techniques of programming through examples using Visual Basic language, from simple to complex applications
- ✓ Experimenting the fundamental algorithms and assimilate the logical thinking steps in the problems resolving is the important paths to develop successful solutions
- ✓ It's important for students to acquire as many visual controls and their properties, events and methods in order to exploit the facilities offered by these in the developed visual interfaces



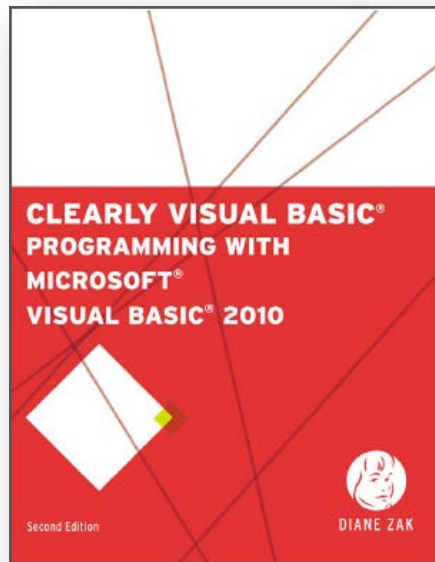
# Computer Programming Course Objectives II



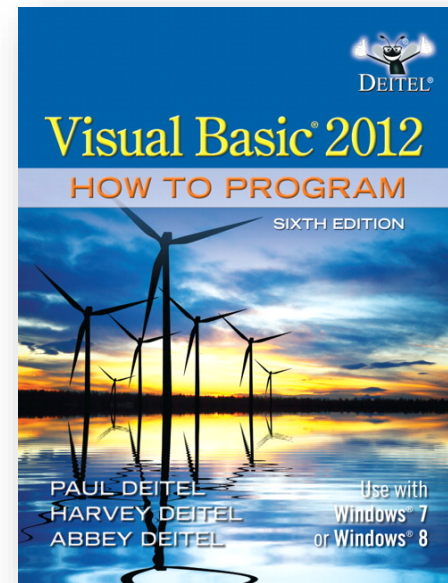
- ✓ The practical activities follow the correct completion of the stages of writing source code, testing, debugging and evaluating the obtained results after program execution
- ✓ The teaching methods are applied through *gamification* in order to attract the students to the beautiful world of programming and to spark the interest in the study of programming languages and developing of the programs
- ✓ It's essential to perform the step-by-step activities in development of source code, from the scratch through simplest to more complex programs, the basic idea is the every time success in writing code and getting the right results, without syntax or run-time errors.

# Overview – Course Structure

- ✓ The Computer Programming has two modules
  - ❑ Course – 4 European Credit Transfer System points
  - ❑ Project – 1 ECTS points
- ✓ Course duration is 14 weeks (2 hours lectures per week, total 28 hours, and 2 hours lab per week, total 28 hours – which is mandatory)
- ✓ Project time: 1 hour per week, total 14 hours.
- ✓ Materials of study consists in courses notes and two recommended textbooks:



and





# Course Structure

- Introduction to Programming
- Creating Applications with Visual Basic
- Data Types, Variables and Calculations
- Decision Structures and Loops
- Procedures and Functions
- Forms, Modules and Menus
- Working with Data Structures: Arrays and Files
- Using Databases in applications
- Classes, Collections and Inheritance



# Lab Content Structure I

- First applications in Visual Basic, from the scratch, very simple operations for familiarizing with writing the first lines of source code
- Creating Windows Form Application and Console Application projects
- Examples of using variables and calculations in economic applications (VAT deduction, economic indicators calculations, inventory management), monitoring the variables at real-time execution of programs
- Different applications using decision structures and loops, students must learn to test the algorithms and debug the source code



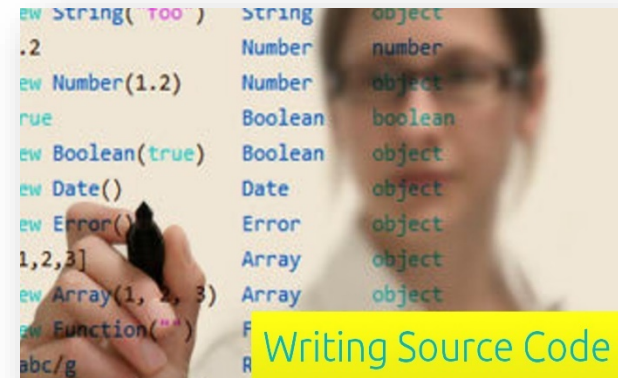
## Lab Content Structure II



- Developing applications with more than one form, using modules and menus to create intuitive and simple to use user interfaces
- Sorting algorithms for numerical and non-numerical values, through different methods (insertion sort, quick sort, bubble sort and variants, bucket sort). Students will compare the efficiency of algorithms and execution times for each method, using the same data set.
- Applications with basic management operations into text files (inserting, deleting, updating, finding and replacing data and sorting records)
- Creating connections to Access or SQL databases and perform records operations, filtering, navigating and visualizing the results

# Teaching Experience Course Lectures I

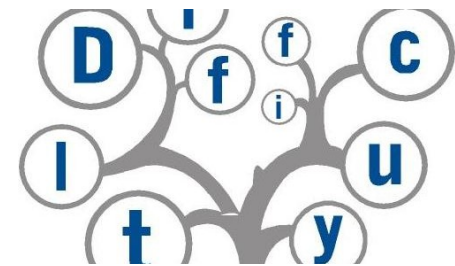
- The power of example is the key to an easy understanding of theoretical notions, each aspect is accompanied, where possible, by a relevant example
- Practical aspects have economic applicability, and are related to lecture elements during the whole course
- The theoretical elements are focused on the main data types and data structures and their use in decision structures and loops.
- It's very important that students understand the logic of the algorithms and focus on complex structures of the applications and highlighting by examples the coding style, providing meaningful names of variables, explicit comments, respect the modularity and an ordered source code



# Teaching Experience Course Lectures II

- The real challenge is to teach the students to work as a team at the same project, to correlate their modules, the input variables and respect the project requirements.

## Difficulties in Educational Experience Lectures



- Students prefer laboratory classes instead lectures - theory versus practical activities ☺. Lectures attendance is less than lab attendance and it is difficult to motivate them to participate at all lecture classes.
- Difficulties arise in the conversion of the pseudocode in the programming language source code, the students tending to treat in the same manner in writing the syntax or forgetting to declare the proper variables
- More complicated algorithms are not fully understood, the students prefer to write directly the source code from the lecture notes in Visual Basic, rather than understand the algorithm and then to write their own source code

# Teaching Experience Laboratories

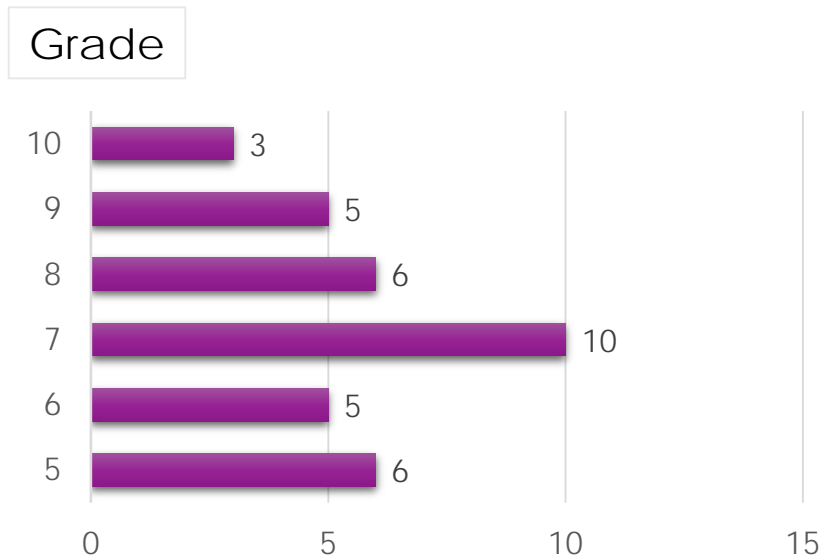
- Homework for each week, proposing students to solve problems similar to those presented into laboratories.
- Students must write the program based on algorithms presented in lecture classes.
- Students must learn to debug their own programs and to correct errors by real-time watching variables and tracking the order execution of the source code lines

## Difficulties in Educational Experience - Laboratories



- Students find that is difficult to follow the logic in solving of problems and forget to have a comprehensive image of solving the problem
- Students tend to write directly the source code and try to repeat running the program and correct the errors instead of analyzing the source code errors and after that running the program.

## Results of the Exam



Grade	No. students
5	6
6	5
7	10
8	6
9	5
10	3
Total passed	35
Failed	5
Absent	3
Total students	43

## Conclusions

- Anyone can program, but not anyone can become a good programmer (note for the students that failed the exam 😊)
- Computer Programming Course creates opportunity for students to learn the basics of programming through an easy way to understand language: Visual Basic
- Step-by-step examples and path from simple to complex it's a good method to assimilate the programming techniques and use programming languages to solve (economic) problems
- Understanding real-life (economic) problems, based on expertise knowledges – leads to correct pseudocode => right code source program
- And not the last, it's necessary that students to spend many hours of work with passion, pay attention and of course, a lot of creation! 😊





**Thanks for your attention!**

