Stat 1129-10  Spring 2011:  Introductory Computer Science

Prof. Robert F. Teitel; email: rteitel@gwu.edu
phone: 301.654.3684
hours: before and after class (please email for appointment)

T/A: Brandon Nedwek; email: bnedwek@gwu.edu
phone: 971.506.7739

Email Note: Please use "STAT129:" at beginning of subject line if sending us email.
Replies may come from rteitel@comcast.net
Replies may come from bnedwek@gmail.com

Class: Mon / Wed  : 5:10 – 6:00 Gov 103 usually lecture
       : 6:10 – 7:00 Gov 103 usually lab

Text: optional:
   a. Williams and Walmsley (out-of-print)
      Discover Delphi: Programming Principles Explained
   b. Barrow, Miller, Malan, Gelderblom
      Introducing Delphi Programming (Oxford/South Africa)

Best: http://www.marcocantu.com/epascal/ , and
      http://www.marcocantu.com/edelphi/
      Free download/pdf/zip files in many (natural) languages.
OR any book on the Pascal / DELPHI system.

A separate guide to internet resources will be distributed.

Class Process:
The sessions in Gov 103 will be lectures and demonstrations for the first few weeks of
the semester; thereafter lecture followed by laboratory time for work on weekly
assignments with the TA available for assistance (often the Professor, too).

Course Content:
This is an introductory overview of computer science using a modern, object-oriented,
visual development environment (Delphi – called the Borland Developer Studio (BDS)). No
previous computer course is assumed, nor is any particular level of mathematics expected.
It is assumed, however, that you are generally familiar with the use of programs (MS Word
/ WordPerfect, email, browsers, games, etc.) on a Windows PC.

We will emphasize very clear distinctions between fundamental compute science concepts
and how those concepts are used in two programming languages, Delphi/Pascal and C++,
though much of the course will use the Delphi/Pascal language.

Software: Borland Developer Studio (BDS2006).
The BDS2006 software incorporates both the Delphi/Pascal and C++ languages, and should be
available in all GWU computing laboratories outside of official class hours. During
official class hours, the BDS2006 software should be available only in Gov 103.

Learning Outcomes:
At the end of the semester the students are expected:
1) to be familiar with the visual design and event driven PC-oriented programming
environment as implemented in BDS2006;
2) to understand the elementary data structures and algorithmic statements common to all
procedural programming languages;
3) to realize how those data structures and algorithmic statements are implemented in the
Delphi/Pascal and C++ programming languages; and
4) to be able to write somewhat useful programs, especially those involving simple
economic and statistical modeling and simulation.
Homework Assignments:
After a few weeks of lectures and demonstrations, each topic will be reinforced by a homework assignment consisting of the development of a practical or interesting program. The specific assignments will depend somewhat on the composition and interest of the class. The assignments will be structured so that at least a part of each one will be due every week. The printed paper version of your program should be submitted – specific instructions will be part of one of the early demonstrations. Sometimes we’ll need to try to execute your program.

You WILL need to work on the assignments OUTSIDE scheduled Laboratory time, especially later in the semester. The assignments will be 50% of the final grade.

You MUST retain backup copies of your programs.
A ZIP disk or a USB disk ("thumb drive") is strongly encouraged (instead of 3.5" disks).

You will generally work in assigned pairs on the weekly programs.
The pairings will vary throughout the semester.
Each pairing submits a single work product.

Assignments will generally be due on Wednesdays BEFORE – or at very start of the – Laboratory time, so that the time can be spent starting on the next assignment with the new assigned pairings.

Graded assignments will usually be returned on Mondays.

Assignments can be re-submitted within a week of their return to you.
An average grade will then be used.

Late assignments will be noted, and will suffer a grade reduction.

Grading:
As noted, assignments are 50% of the final grade.
A document will be distributed indicating the grading process used for the assignments.

There will be one midterm exam which will count 20% toward the final grade.

The cumulative final exam will count 30% toward the final grade.

The midterm and the final will be OPEN book and OPEN notes (but no collaboration!)

Academic Integrity Issues:
For the weekly assignments, you will generally work in pairs. You may also use printed and electronic (internet) resources, but you must indicate that you have done so as comments in the programs (just as you would footnote such references in conventional papers). You may even have incidental conversations with fellow students; helpful insights should, once again, be indicated as comments in the programs. Communication with, or help from, persons not in this class is not permitted. Regardless of the resources used, the work submitted must still represent your own work product.

For the midterm and the final exams, the above still hold, except that you may not have any communication with any person, whether in this class or not. That is, the exam work products shall be completely your own with, of course, properly attributed researched resources.

Any work you submit will be held to these standards. Failure to do so will be appropriately sanctioned.
Assignments: Specific assignments depend somewhat on the makeup of the class. If you have a particular interest in some topic let me know.

After a brief introduction to the history of modern computing, the first couple of weeks will be spent on becoming acquainted with “Visual design”, “Object properties”, and “Event handlers”.

As you will appreciate during the semester, basic computer science consists of learning about “data structures” and about “algorithms (and algorithmic statements)”. Lectures and assignments will oscillate between the two in roughly the following order:

<table>
<thead>
<tr>
<th>Data structures</th>
<th>Algorithm construction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic scalar data types</td>
<td>Assignment statement; expressions</td>
</tr>
<tr>
<td>(Integer, String, Enumerated, Boolean)</td>
<td>Conditional execution</td>
</tr>
<tr>
<td>Char data type</td>
<td>(IF - THEN - ELSE)</td>
</tr>
<tr>
<td>Real data type</td>
<td>Count-controlled iteration</td>
</tr>
<tr>
<td></td>
<td>(FOR - DO)</td>
</tr>
<tr>
<td>Arrays</td>
<td>Conditional iteration</td>
</tr>
<tr>
<td>Recursion</td>
<td>(WHILE - DO)</td>
</tr>
<tr>
<td></td>
<td>(REPEAT - UNTIL)</td>
</tr>
<tr>
<td>Functions and procedures</td>
<td>Conditional execution</td>
</tr>
<tr>
<td></td>
<td>(CASE - OF)</td>
</tr>
</tbody>
</table>