CIS-262 Computer Organization and Design
Bristol Community College
Computer Information Systems Department
Spring 2020

Catalog Description

Laws of computer organization and design for RISC architectures. Interfaces between hardware and software are studied. Influence of instruction set on performance is presented. Design of a processor with pipelining is analyzed. Computer arithmetic is studied. Memory hierarchy and their influence on performance are documented. Elements of interfacing and I/O organization are included. The course has design, implementation, and analytical components.

Three class hours and two lab hours per week.

Meeting days and times:

Lecture: room K-205 Tuesdays 4:00 pm – 6:45 pm.

Laboratory: room K-105 Thursdays 5:00 pm – 6:15 pm.

Prerequisite:

CIS 261 (formerly CIS 77) or permission of the instructor.

- Instructor: Igor Kholodov `Igor.Kholodov@bristolcc.edu`
- Office: K211

- Telephone: `774-357-3328`

Student Learning Outcomes

The course will give students a better understanding of computing that will lay a foundation for more advanced course. At the completion of this course the student will be able to:

- Explain how a program written in a high level language such as C or Java is translated into the language of the hardware
- Explain how hardware executes programs that have been translated

- Determine the interface between hardware and software

- Describe how software instructs hardware to perform a certain task

- Understand how hardware design determines the performance of a program

Specific goals to meet these outcomes include:

- Understanding how program algorithms can determine the performance of a program
- Defining which programming algorithms will improve performance on given hardware

- Programming highly parallel, highly multithreaded multiprocessor optimized for visual computing

Recommended Texts

Title: Computer Organization and Design
by David A. Patterson and John L. Hennessy
Publisher: Morgan Kaufmann
5th edition
ISBN-10: 0124077269
ISBN-13: 978-0124077263

Disability Accommodations

I encourage any student in need of accommodations for a specific documented disability to meet with me and the Office of Disability Services (508) 678-2811 (Fall River, ext. 2955; Attleboro and Taunton, ext. 2996; New Bedford, ext. 2955 and/or 4011) at your earliest convenience to ensure timely and appropriate accommodations. You may also contact the Office of Disability Services (ODS) online at
http://www.bristolcc.edu/about/publicrecordsrequest/disabilityservices/

Course Organization

- Two major components of this course are **mid-** and **end-semester** projects. Both can be developed in groups (2-3 students) or individually.
- Each group will give a short (10-15 minute) presentation of the end-semester project at the end of the course.

- There will be weekly programming laboratories - attendance in labs is mandatory. The labs will start the second week of classes. The labs are designed to help you with the semester projects. In the laboratories you will practice programming techniques used in multiprocessor-optimized visual computing.

Minimum Requirements for a Passing Grade

- Curiosity and love for learning.
- Responsibility for reading the textbook and keeping current with the class material.

- Ability to research any subject not covered in text or study guides.

- Average grade of 60 or greater on Midterm and Endterm exams.

- Completion of all design stages of the **mid-** and **end-semester** projects.

Weights for the Final Grade Determination

```
Labs/Attendance       60%
Mid-semester Project  20%
End-semester Project  20%
```

The Final Grades will be assigned as follows:

```
   97-100 A+     87-89  B+     77-79  C+     67-69  D+
   93-96  A      83-86  B      73-76  C      63-66  D
   90-92  A-     80-82  B-     70-72  C-     60-62  D-
Below 60  F
```

Teaching Methodology

The lecture will be the principal teaching method used in this course. "Handouts" and sample programs will be available on the class web page. Class discussions will be conducted pertaining to the lab exercises. Software demos and overhead slides will be used.

It will be imperative that the student complete all assigned readings and homework assignments *prior to class*. Failure to do so is a formula for failure. Coming to class prepared is essential for successful completion of the course.

Attendance Policy

Attendance is recorded weekly based on the student's ability to complete quality and timely lab/programming assignments each week. Students are considered "present" for the week if they submit the required lab assignment (with a satisfying passing grade) prior to the due date for that week. Poor attendance may affect your final grade.

Students are responsible for withdrawing officially if they stop attending any or all classes. Faculty no longer have the ability to withdraw a student from a class. A grade of "F" will be assigned to any student who stops attending a course but does not officially withdraw. Students are encouraged to meet with an advisor before making any changes to their schedule. Withdrawals impact Satisfactory Academic Progress and can place the student at risk for academic probation or dismissal. Students who use financial aid and who subsequently withdraw may be required to return some or all funds received. Withdrawals are accepted until the tenth week of classes. Students may withdraw online in accessBCC, in person at any Enrollment Center, or via their college email to enrollmentservices@bristolcc.edu. Email requests must come from the student's BCC college email address and must include the student's name, BCC student ID number, and course information (CRN, course and section number).

Email from non-college accounts will not be accepted. If a student officially withdraws after the third week of classes, there will be no tuition or college fee refunds. For more information, see the College Catalog at

http://bristolcc.edu/

Students with questions should contact Enrollment Services via any of the methods mentioned above or at 774-357-2590.

BCC Academic Policies

College-wide Academic Policies outlined in BCC Academic Catalog directly apply to this course. It is your responsibility to read carefully and understand Academic Information, especially Academic Integrity, Academic Dishonesty, Academic Negligence, Plagiarism, and Classroom conduct, which are published online.

CIS-262 TENTATIVE SCHEDULE

Week 1. Intro to Computer Organization and Design

- Reading: BCC Student Handbook, page 8, Tip #5:
    - *"For each hour in class, you should expect to study **at least 2-3 hours outside of class**. Know your limits, avoid over scheduling yourself (whether it be work or class). Set up a schedule that you know will allow you to earn good grades. And, maintain a day planner to help you stay organized."*

- CIS-261, Intro to Computer Systems, and CIS-158, Intro to Procedural Programming, skills review: Von Neumann model -- Language of the computer -- Binary and Integer arithmetic, C/C++ Memory Access

Week 2. Assembler, Compiler, Linker, and Loader

- Source, library, and object files
- Executable program images and Dynamic Link Libraries (DLLs)

- OS Loader

- Introduction to OpenGL

- Lab 1: Visual Studio, OpenGL, DLLs, and LIBs

Week 3. MIPS Assembly

- Operation and operands of the computer hardware
- MIPS instructions and operands

- Logical instructions for making decisions

- Procedures and procedure calls

- MIPS addressing modes

- OpenGL Geometry Primitives

Week s 4-5. Floating Point Data Type

- Floating Point Primitive Types in C/C++
- OpenGL Shaders Intro

- Example: Shaders Demo

- Lab 2: Compiling and Linking Shader Programs

Week 6. The Processor Performance

- CPU Datapath
- Control Unit

- GPU Pipelining

- Lab 2 follow-up and solution

- Example: Vertex Array Objects and Vertex Buffer Objects

- Lab 3: Vertex Array Objects and Vertex Buffer Objects

Week 7. Floating Point Arithmetic

- Example: Index Buffers
- Demo program: transformations to move the camera around the scene. Keyboard controls: Esc/q=quit W=zoom-in S=zoom-out a/d-left-right Y/y-up-down e/r-yaw (rotate around the camera left/right)

- Trigonometry, Vectors, Matrices, GLM

Week 8. Memory Hierarchy

- Memory Caches
- Cache Performance

- Virtual Memory

- Memory Heirarchies

- Disk Storage

- Lab 5: OpenGL Index Buffers and Animation

- Mid-semester project: battleship silhouette

_____*Spring Break*_____

Week 9. GPU Architectures

- Programmable Pipeline and GLSL
- Lab 6: Wavefront OBJ files

Week 10. Lighting Models in OpenGL

- Diffuse and Specular Light
- Example: diffuse light (ZIP)

- Example: diffuse lighting using OBJ loader (ZIP)

- Lab 7: Diffuse and specular light

Week 11. Graphics and Computing GPUs

- Example: specular light
- Example: specular lighting using OBJ loader

Week 12. Keyboard and mouse

- OpenGL Application Structure Overview
- OpenGL Application Scaffolding Notes

- Lab 8: Keyboard and mouse events with GLUT

Week 13. Textures

- Texture Mapping
- Example: Texture

- Lab 8: follow-up and solution

- Lab 8 solution with second model (horizontal plane) added

- Lab 9: Texture Mapping

Week 14. Texture cube mapping

- Cube mapping
- Lab 10: Skyboxes

Week 15. End-semester project review

- End-semester project-related activity all week

End-semester Presentations

- End-semester project is due
- Project review sessions, strictly by schedule.

Note: *This syllabus is a suggested course outline and will be generally followed, subject to change according to the instructor's discretion and needs. Academic flexibility is important.*