Course Change Request

New Course Proposal

Date Submitted: 02/14/24 9:29 pm

Viewing: CDS 351 : Elements of High Performance Computing

Last edit: 02/14/24 9:29 pm

Changes proposed by: blaisten

Are you completing this form on someone else's behalf?

No

In Workflow

1. **CDS Chair**
2. **SC Curriculum Committee**
3. SC Assistant Dean
4. Assoc Provost-Undergraduate
5. Registrar-Courses
6. Banner

Approval Path

1. 02/15/24 10:41 am Jason Kinser (jkinser): Approved for CDS Chair

Effective Term:	Fall 2024		
Subject Code:	CDS - Computational and Data Sciences	**Course Number:**	351
Bundled Courses:			
Is this course replacing another course?	No		
Equivalent Courses:			
Catalog Title:	Elements of High Performance Computing		
Banner Title:	High Performance Computing		
Will section titles vary by semester?	No		
Credits:	3		
Schedule Type:	Lecture		
Hours of Lecture or Seminar per week:	3		
Repeatable:	May only be taken once for credit, limited to 2 attempts (N2)	**Max Allowable Credits:**	

3

Default Grade Mode: Undergraduate Regular

Recommended Prerequisite(s):

Recommended Corequisite(s):

Required Prerequisite(s) / Corequisite(s) (Updates only):

CDS 251 or equivalent, or permission of the instructor

Registrar's Office Use Only - Required Prerequisite(s)/Corequisite(s):

And/Or	(	Course/Test Code	Min Grade/Score	Academic Level	)	Concurrency?

Registration Restrictions (Updates only):

Registrar's Office Use Only - Registration Restrictions:

Field(s) of Study:

Class(es):

Level(s):

Degree(s):

School(s):

Catalog Description:

The course explores aspects of high-performance computing (HPC) based on a diverse set of tools, including Unix basics, file systems, command scripts, Git, C++ programming, basics of parallel programming, and HPC system architectures.

Justification:

What: The course aims to empower students with a robust skill set and knowledge base ensuring they are well-prepared for the challenges and opportunities presented by the dynamic field of high-performance computing.

Why: The course content establishes a foundation in essential technologies and methodologies used in high-performance computing (HPC), preparing students with a skill set vital for exploiting the potential of modern computing architectures required for more advanced coursework and research opportunities.

Does this course cover material which crosses into another department? No

Learning Outcomes:

Upon completion of the course, students will attain a comprehensive skill set essential in cutting-edge computational environments. They will be proficient on fhe fundamentals of Unix-based operating systems, command line navigation, file and directory management, and utilizing key Unix tools tailored for HPC settings. Additionally, students will gain competence in software development and build systems to ensure the efficient and error-free deployment of HPC applications. The course will introduce the basic elements of C++ programming, emphasizing its application in solving computational challenges pertinent to HPC. Students will delve into the basics of HPC system architectures that constitute high-performance computing systems. Furthermore, students will acquire basic proficiency in parallel programming involving the utilization of multi-core processors and distributed computing environments, empowering them with parallelization techniques for performing complex computational tasks.

Will this course be scheduled as a cross-level cross listed section? No

Attach Syllabus
Syllabus_Elements_of_HPC_Feb14.pdf

Additional Attachments

Staffing:
Name of the potential instructors within the Department of Computational and Data Sciences:
Dr. Sharmin Abdullah (Instructional Assistant Professor, full time)
Dr. Carlos Cruz (Assistant Adjunct Professor)

Relationship to Existing Programs:
None

Relationship to Existing Courses:
Currently, Dr. Abdullah is teaching CDS 130 (Computing for Scientists) and Dr. Cruz is teaching CDS 230 (Modeling and Simulation I).

Additional Comments:

Reviewer Comments

Key: 18575

CDS 351: Elements of High Performance Computing

Time: TBD
Location: TBD
Course website: TBD

Instructor: Drs. Sharmin Abdullah or Carlos Cruz
Office hours: TBD

Prerequisites: CDS 251 and knowledge of one operating system, or permission of the instructor

Course description: This course is designed to prepare you for the dynamic world of high-performance computing (HPC). The course covers HPC essential foundations, including Unix basics, file systems, command scripts, Git, C++ programming (key language for HPC development), the basics of parallel programming, and exploration of HPC system architectures.

Course expectations: Students are expected to actively engage in class discussions, group activities, and assignments. This includes asking questions, sharing insights, and collaborating with peers. Regular attendance is crucial. Students are expected to come to class prepared. This involves completing assigned readings, reviewing lecture notes, and being ready to participate in discussions and activities.

Course learning outcomes:

- Basic proficiency in Unix-based operating systems by effectively navigating the command line, managing files and directories, and utilizing essential Unix tools for HPC environments.
- Competence in building and compiling software by creating, configuring, and maintaining build systems, ensuring the development of efficient and error-free HPC applications.
- Applying C++ programming concepts for solving computational problems relevant to HPC.
- Identifying HPC system architectures by explaining the components, designs, and topologies used in high-performance computing systems.
- Basic proficiency in parallel programming by designing and implementing parallel algorithms, effectively using multi-core processors and distributed computing environments to interpret the use of such techniques for solving complex computational tasks.

Students will apply their knowledge in hands-on projects that involve the development and optimization of HPC applications, providing practical experience in real-world HPC challenges.

Computer Resources: The CDS computer lab and with 20 Linux workstation will be used. Other computational resources may be used as they become available. Students will access physically this lab or use their personal computers (laptops or desktops) to access these workstations.

Recommended References:

- Beginner C++: Programming: Principles and Practice Using C++ (2nd Edition), Bjarne Stroustrup
- Intermediate C++: The C++ Programming Language, Fourth Edition, Bjarne Stroustrup
- Introduction to High-Performance Computing for Scientists and Engineers (Chapman & Hall/CRC Computational Science), Georg Hager, Gerhard Wellein
- Introduction to High Performance Computing, (evolving online version), Victor Eijkhout, Edmond Chow, Robert van de Geijn

Grading and course requirements:
Homework will consist of conceptual exercises, programming assignments, as well as group projects. There will be one mid-term examination and a final exam.

Conceptual exercises and programming assignments:

- Basic Unix (shell and make)
- C++ programming exercises including
  - Creating applications and automation
  - Timing and performance measures
- HPC concepts
- Parallel programming group exercises. Students will work in small groups to solve representative two or three HPC related exercises chosen from the list below:
  - Data processing of large datasets
  - Parallel image processing
  - Parallel sorting
  - Parallel summation and matrix multiplication
  - Genetic algorithms
  - Monte Carlo integration

Grade weights:

Homework	60%
Midterm	20%
Final	20%
Total	100%

Grade	Total
A+	> 96.7
A	93.3-96.6
A-	90.0-93.2
B+	86.7-89.9
B	83.3-86.6
B-	80.0-83.2
C+	76.7-79.9
C	73.3-76.6
C-	70.0-73.2
D	60.0-69.9
F	< 60.0

Tentative course schedule:

Prerequisites
Week 1 ..... Introduction, Unix shell
Week 2 ..... File systems, build systems
Week 3 ..... Git, IDEs
Programming
Week 4 ..... Introduction to C++ (variables, functions)
Week 5 ..... Control Flow, Arrays
Week 6 ..... File IO, Introduction to OOP
Week 7 ..... **Midterm Exam**
Distributed programming
Week 8 ..... Introduction to HPC
Week 9 ..... Clusters, SSH
Week 10 ..... Memory architectures
Week 11 ..... Thread Parallelism, OpenMP
Week 12 ..... Message Passing Interface (MPI)
Week 13 ..... Applications
Week 14 ..... Applications
Week 15 ..... **Final Exam**

Mason Honor Code (quote from catalog.gmu.edu): “To promote a stronger sense of mutual responsibility, respect, trust, and fairness among all members of the George Mason University community and with the desire for greater academic and personal achievement, we, the student members of the university community, have set forth this honor code: Student members of the George Mason University community pledge not to cheat, plagiarize, steal, or lie in matters related to academic work.”

Email policy: Mason’s electronic mail provides any official information to students. Any class materials, assignments, questions, and instructor feedback should use the email Mason email. Students are responsible for maintaining their email account active, working correctly, and should check their content regularly (review details in catalog.gmu.edu).

The course AI policy: Use of Artificial Intelligence (AI) tools (Chat GPT, etc.) may enhance student learning. It is to be noted that academic instructors are able to identify AI-generated work with 94% accuracy. Additionally, Mason utilizes special techniques to scan assignments for detecting cheating, including AI generated materials. In this course AI tools may only be used to support the student learning. Use of AI tools on the course assignments outside of areas specifically approved by the instructor constitute cheating. Any form of cheating on an activity, project, or exam will result in zero points earned.

Plagiarism policy for Internet materials: Copyright rules apply to users of the Internet who employ elements downloaded from Internet sources. Any information in the form of graphics, text, tables, or data accessed electronically and used in homework, presentations, exams, email, reports, must be cited giving credit to the pertaining sources. Even if credit is given, students must obtain permission from any copyrighted source to use any material not created by them. Inserting someone's else material in your work is stealing intellectual property. Including a link to the site URL is currently an appropriate citation.

Student privacy policy: Mason complies with FERPA by protecting the privacy of student records and judiciously evaluating requests for release of information from those records. It is not permitted for faculty to share class progress or grade information with parents/guardians under any circumstances. Student privacy policy: https://registrar.gmu.edu/students/privacy/

Academic integrity: This course embodies the value that we all have differing perspectives and ideas, and we each deserve the opportunity to share our thoughts. Therefore, we will conduct our discussions with respect for those differences. That means, we each have the freedom to express our ideas, but we should also do so keeping in mind that our colleagues deserve to hear differing thoughts in a respectful manner, i.e. we may disagree without being disagreeable. https://oai.gmu.edu/

Students with disabilities: Students with disabilities should contact the Office of Disability Services (ODS). Students requiring special accommodations should inform the instructor the first week of classes. Accommodations may be appropriate for situations that directly affect the student academic performance. ODS requires pertinent medical documentation of a physical, mental health, attention, or other health challenge. https://ds.gmu.edu/.