New York City College of Technology, CUNY

CURRICULUM MODIFICATION PROPOSAL FORM

This form is used for all curriculum modification proposals. See the [Proposal Classification Chart](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/2013-10-09-Proposal_Classification_Chart.pdf) for information about what types of modifications are major or minor. Completed proposals should be emailed to the Curriculum Committee chair.

|  |  |
| --- | --- |
| **Title of Proposal** | **CST 4723 and CST 3650** |
| **Date** | **7/17/17** |
| **Major or Minor** | **Major** |
| **Proposer’s Name** | **Candido Cabo and David Bellehsen** |
| **Department** | **Computer Systems Technology** |
| **Date of Departmental Meeting in which proposal was approved** | **5/5/17** |
| **Department Chair Name** | **Hong Li** |
| **Department Chair Signature and Date** |  |
| **Academic Dean Name** | **Kevin Hom** |
| **Academic Dean Signature and Date** |  |
| **Brief Description of Proposal**(Describe the modifications contained within this proposal in a succinct summary. More detailed content will be provided in the proposal body. | **Two new major elective courses:****1) CST 4723-High Performance Computing and Parallel Programming****2) CST 3650 – Data Structures** |
| **Brief Rationale for Proposal**(Provide a concise summary of why this proposed change is important to the department. More detailed content will be provided in the proposal body).  | **These courses will allow students to further develop their programming skills and knowledge.** |
| **Proposal History**(Please provide history of this proposal: is this a resubmission? An updated version? This may most easily be expressed as a list). | **First submission: 9/15/2017****Second revision: 10/11/2017****Third revision: 10/19/2017****This is the fourth revision: 11/12/17** |

Please include all appropriate documentation as indicated in the Curriculum Modification Checklist.

For each new course, please also complete the New Course Proposal and submit in this document.

Please submit this document as a single .doc or .rtf format. If some documents are unable to be converted to .doc, then please provide all documents archived into a single .zip file.

**ALL PROPOSAL CHECK LIST**

|  |  |
| --- | --- |
| Completed CURRICULUM MODIFICATION FORM including: |  |
| * Brief description of proposal
 | x |
| * Rationale for proposal
 | x |
| * Date of department meeting approving the modification
 | x |
| * Chair’s Signature
 | x |
| * Dean’s Signature
 | x |
| Evidence of consultation with affected departmentsList of the programs that use this course as required or elective, and courses that use this as a prerequisite. | x |
| Documentation of Advisory Commission views (if applicable). |  |
| Completed [Chancellor’s Report Form](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/2013-10-09-Chancellor_Report_Quick_Reference_Guide1.doc). | x |

**EXISTING PROGRAM MODIFICATION PROPOSALS**

|  |  |
| --- | --- |
| Documentation indicating core curriculum requirements have been met for new programs/options or program changes.  |  |
| Detailed rationale for each modification (this includes minor modifications) |  |

**Chancellor’s Report**

**Section AIV: New Courses**

**AIV.1. Department:** Computer Systems Technology

**New courses to be offered in the Computer Systems Technology department**

|  |  |
| --- | --- |
| **Department(s)** | Computer Systems Technology |
| **Academic Level** | **[ x  ] Regular  [   ] Compensatory  [   ] Developmental  [   ] Remedial**  |
| **Subject Area** | Computer Systems |
| **Course Prefix** | CST |
| **Course Number** | 4723 |
| **Course Title** | High Performance Computing and Parallel Programming |
| **Catalog Description** | Fundamentals of high performance computing and parallel programming. Introduces different computer architectures used for parallel computations such as multicore processor systems, computer clusters and graphic processing units (GPU). Develops skills such as setting up and configuring a computer system to run programs concurrently, and programming parallel architectures using the appropriate application programming interfaces (MPI, OpenMP, OpenCL/CUDA). Covers strategies to parallelize and optimize computer programs such as profiling, partitioning and load balancing. Common applications of parallel programming are discussed. |
| **Prerequisite** | CST3503 or CST3513 or CST3523 |
| **Corequisite** |  |
| **Pre- or corequisite** |  |
| **Credits** | 3 |
| **Contact Hours** | 4 |
| **Liberal Arts** | **[ ] Yes  [ x  ] No**  |
| **Course Attribute (e.g. Writing Intensive, etc)** |  |
| **Course Applicability** |

|  |  |
| --- | --- |
| **[x ] Major** |  |
| **[ ] Gen Ed Required** | **[ ] Gen Ed – Flexible** | **[ ] Gen Ed - College Option** |
| **[ ] English Composition** | **[ ] World Cultures** | **[ ] Speech** |
| **[ ] Mathematics** | **[ ] US Experience in its Diversity** | **[ ] Interdisciplinary** |
| **[ ] Science** | **[ ] Creative Expression** |  **[ ] Advanced Liberal Arts** |
|  | **[ ] Individual and Society** |  |
|  | **[ ] Scientific World** |  |

 |
| **Effective Term** | Fall 2018 |

**Rationale:** This course will allow students to further develop their programming skills and knowledge by learning about strategies to parallelize and optimize computer programs.

**New courses to be offered in the Computer Systems Technology department**

|  |  |
| --- | --- |
| **Department(s)** | Computer Systems Technology |
| **Academic Level** | **[ x  ] Regular  [   ] Compensatory  [   ] Developmental  [   ] Remedial**  |
| **Subject Area** | Computer Systems |
| **Course Prefix** | CST |
| **Course Number** | 3650 |
| **Course Title** | Data Structures |
| **Catalog Description** | Introduces structure of data in computer memory, including arrays, linked lists, stacks, trees, hash tables, graphs.  Discuss algorithms to manipulate data in these structures in various ways, such as searching for a data item and sorting a set of data elements.  Covers algorithms such as how to organize data elements, how to delete, insert, edit and search for a data element in a specific data structure, how to sort a set of data elements and the differences between different data structures and when to use the right ones to solve problems. |
| **Prerequisite** | CST3503 or CST3513  |
| **Corequisite** |  |
| **Pre- or corequisite** |  |
| **Credits** | 3 |
| **Contact Hours** | 4 |
| **Liberal Arts** | **[ ] Yes  [ x  ] No**  |
| **Course Attribute (e.g. Writing Intensive, etc)** |  |
| **Course Applicability** |

|  |  |
| --- | --- |
| **[x ] Major** |  |
| **[ ] Gen Ed Required** | **[ ] Gen Ed – Flexible** | **[ ] Gen Ed - College Option** |
| **[ ] English Composition** | **[ ] World Cultures** | **[ ] Speech** |
| **[ ] Mathematics** | **[ ] US Experience in its Diversity** | **[ ] Interdisciplinary** |
| **[ ] Science** | **[ ] Creative Expression** |  **[ ] Advanced Liberal Arts** |
|  | **[ ] Individual and Society** |  |
|  | **[ ] Scientific World** |  |

 |
| **Effective Term** | Fall 2018 |

**Rationale:** **:** This course will add to current BTech elective options and allow students to further develop programming skills and knowledge, especially for student who might seek continuous study in Computer Science

New York City College of Technology, CUNY

NEW COURSE PROPOSAL FORM

This form is used for all new course proposals. Attach this to the [Curriculum Modification Proposal Form](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/2013-10-10-Curriculum_Modification_Proposal_Form.docx) and submit as one package as per instructions. Use one New Course Proposal Form for each new course.

|  |  |
| --- | --- |
| **Course Title** | **High Performance Computing and Parallel Programming** |
| **Proposal Date** | 7/17/17 |
| **Proposer’s Name**  | Candido Cabo |
| **Course Number** | **CST 4723** |
| **Course Credits, Hours** | 3 credits, 4 hours |
| **Course Pre / Co-Requisites** | CST3503 or CST 3513 or CST 3523 |
| **Catalog Course Description** | Fundamentals of high performance computing and parallel programming. Introduces different computer architectures used for parallel computations such as multicore processor systems, computer clusters and graphic processing units (GPU). Develops skills such as setting up and configuring a computer system to run programs concurrently, and programming parallel architectures using the appropriate application programming interfaces (MPI, OpenMP, OpenCL/CUDA). Covers strategies to parallelize and optimize computer programs such as profiling, partitioning and load balancing. Common applications of parallel programming are discussed. |
| **Brief Rationale**Provide a concise summary of why this course is important to the department, school or college. | With most computers shipping now with multiple cores, it is important to introduce students to write computer programs that can run in multiple cores and multiple computers. Computationally intensive programs benefit from parallelizing computer code. Computationally intensive programs are common in scientific computing and data science. |
| **CUNY – Course Equivalencies**Provide information about equivalent courses within CUNY, if any. | There are similar courses in other CUNY colleges:* CSC 229 – Introduction to High Performance Computing (College of Staten Island)
* CSCI 344 - Distributed Systems **(**Queens College)
* CISC 3330 - Foundations of Parallel and Distributed Computing (Brooklyn College)
* CSC 43000 Distributed Computing (City College)
 |
| **Intent to Submit as Common Core**If this course is intended to fulfill one of the requirements in the common core, then indicate which area. | No |
| **For Interdisciplinary Courses:*** Date submitted to ID Committee for review
* Date ID recommendation received

- Will all sections be offered as ID? Y/N | No |
|  |
|  |
| **Intent to Submit as a Writing Intensive Course** | No |

Please include all appropriate documentation as indicated in the NEW COURSE PROPOSAL Combine all information into a single document that is included in the Curriculum Modification Form.

**NEW COURSE PROPOSAL CHECK LIST**

Use this checklist to ensure that all required documentation has been included. You may wish to use this checklist as a table of contents within the new course proposal.

|  |  |
| --- | --- |
| **Completed NEW COURSE PROPOSAL FORM** |  |
| * Title, Number, Credits, Hours, Catalog course description
 | X |
| * Brief Rationale
 | X |
| * CUNY – Course Equivalencies
 | X |
| Completed [Library Resources and Information Literacy Form](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/curriculum_modification_library_form.doc) |  |
| **Course Outline** Include within the outline the following. |  |
| Hours and Credits for Lecture and LabsIf hours exceed mandated Carnegie Hours, then rationale for this | X |
| Prerequisites/Co- requisites | X |
| Detailed Course Description | X |
| Course Specific Learning Outcome and Assessment Tables* Discipline Specific
* General Education Specific Learning Outcome and Assessment Tables
 | X |
| Example Weekly Course outline | X |
| Grade Policy and Procedure | X |
| Recommended Instructional Materials (Textbooks, lab supplies, etc) | X |
| Library resources and bibliography | X |
| **Course Need Assessment.** Describe the need for this course. Include in your statement the following information. |  |
| Target Students who will take this course. Which programs or departments, and how many anticipated?Documentation of student views (if applicable, e.g. non-required elective). | X |
| Projected headcounts (fall/spring and day/evening) for each new or modified course. | X |
| If additional physical resources are required (new space, modifications, equipment), description of these requirements. If applicable, Memo or email from the VP for Finance and Administration with written comments regarding additional and/or new facilities, renovations or construction. | X |
| Where does this course overlap with other courses, both within and outside of the department? | X |
| Does the Department currently have full time faculty qualified to teach this course? If not, then what plans are there to cover this? | X |
| If needs assessment states that this course is required by an accrediting body, then provide documentation indicating that need. |  |
| **Course Design**Describe how this course is designed.  |  |
| Course Context (e.g. required, elective, capstone) | X |
| Course Structure: how the course will be offered (e.g. lecture, seminar, tutorial, fieldtrip)? | X |
| Anticipated pedagogical strategies and instructional design (e.g. Group Work, Case Study, Team Project, Lecture) | X |
| How does this course support Programmatic Learning Outcomes? | X |
| Is this course designed to be partially or fully online? If so, describe how this benefits students and/or program. | X |
| **Additional Forms for Specific Course Categories** | NA |
|  [Interdisciplinary Form](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/Application-for-Interdisciplinary-Course-Designation.docx) (if applicable) |  |
|  Interdisciplinary Committee Recommendation (if applicable and if received)\* \*Recommendation must be received before consideration by full Curriculum Committee |  |
| [Common Core (Liberal Arts) Intent to Submit](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/CommonCoreCourseSubmissionForm_4.2.12.doc) (if applicable) |  |
| Writing Intensive Form if course is intended to be a WIC (under development)  |  |
| If course originated as an experimental course, then results of evaluation plan as developed with director of assessment. |  |
| **(Additional materials for** [**Curricular Experiments**](http://www.300jaystreet.com/college-council/curriculum_proposals/curricular-experiments)**)** | NA |
| Plan and process for evaluation developed in consultation with the director of assessment. (Contact Director of Assessment for more information). |  |
| Established Timeline for Curricular Experiment |  |

**Course Need Assessment:**

This course will be an elective for Computer Systems Technology students. Students in Applied Mathematics and Applied Computational Physics may also benefit from this course.

One section of the course will be offered in fall and spring semesters. Expected student count is 24 students per semester.

**No additional space is necessary. The course can be run at the facilities of the CUNY High Performance Computer Center located at the College of Staten Island (see below).**

No overlap with existing Computer Systems Technology courses. Minimal overlap with PHYS 4100 (Computational Methods). While the main topics of PHYS 4100 are the application of numerical methods to problems in physics, it also includes some basics of high performance computing. The approach in this proposed course is not on numerical methods applied to problems in physics, but on the general computational and technical aspects of high performance computing and parallel programming.

The department has faculty members who are capable of teaching this course.

**Course Design:**

This course will be an elective for Computer Systems Technology students. Students in Applied Mathematics and Applied Computational Physics may also benefit from this course.

The course will be a combination of theory and hands-on (2 hours lecture; 2 hours lab), including the writing of short computer programs to understand basic concepts and a larger group project.

**LIBRARY RESOURCES & INFORMATION LITERACY: MAJOR CURRICULUM MODIFICATION**

Please complete for **all** major curriculum modifications. This information will assist the library in planning for new courses/programs.

Consult with your library faculty subject specialist (<http://cityte.ch/dir>) **3 weeks before the proposal deadline**.

**Course proposer:** please complete boxes 1-4. **Library faculty subject specialist:** please complete box 5.

|  |  |  |
| --- | --- | --- |
| **1** | **Title of proposal****CST 4723 – High Performance Computing and Parallel Programming** | **Department/Program**Computer Systems TechnologyB.Tech. in Computer Systems |
|  | **Proposed by** (include email & phone)Candido CaboPhone: (718) 260-5162Email: ccabo@citytech.cuny.edu | **Expected date course(s) will be offered** Fall 2018**# of students 24** |

|  |  |
| --- | --- |
| **2** | **The library cannot purchase reserve textbooks for every course at the college, nor copies for all students. Consult our website (**[**http://cityte.ch/curriculum**](http://cityte.ch/curriculum)**) for articles and ebooks for your courses, or our open educational resources (OER) guide (**[**http://cityte.ch/oer**](http://cityte.ch/oer)**). Have you considered using a freely-available OER or an open textbook in this course?**Yes.  |

|  |  |
| --- | --- |
| **3** | **Beyond the required course materials, are City Tech library resources sufficient for course assignments? If additional resources are needed, please provide format details (e.g. ebook, journal, DVD, etc.), full citation (author, title, publisher, edition, date), price, and product link.**No additional resources are required beyond the required course materials.  |

|  |  |
| --- | --- |
| **4** | **Library faculty focus on strengthening students' information literacy skills in finding, critically evaluating, and ethically using information. We collaborate on developing assignments and customized instruction and research guides. When this course is offered, how do you plan to consult with the library faculty subject specialist for your area? Please elaborate.**We may consult with the library faculty specialist on an ad hoc basis, but we do not anticipate a semester long need for library faculty consultation.  |

|  |  |
| --- | --- |
| **5** | **Library Faculty Subject Specialist Prof. Junior Tidal****Comments and Recommendations**After surveying the library’s holdings, I suggest that we acquire more monographs related to cluster computing, parallel programming, and GPUs other than the required texts. Although we have numerous eBooks related to cluster computing and parallel programming, the library’s physical collection could certainly be augmented to further support the students in this course.**Date 09.12.17** |

**NEW YORK CITY COLLEGE OF TECHNOLOGY/CUNY**

**Computer Systems Technology Department**

**COURSE: CST 4723 – High Performance Computing and Parallel Programming**

**(2 class hours; 2 lab hours; 3 credits)**

**Course Description:**

This course covers the fundamentals of high performance computing and parallel programming. Different computer architectures that can be used for parallel computations like multicore processor systems, computer clusters and graphic processing units (GPU), are introduced. Student learn how to set up and configure a computer system to run programs concurrently and then learn to program those parallel architectures using the appropriate application programming interfaces (MPI, OpenMP, OpenCL/CUDA) through programming exercises. The course also covers strategies to parallelize and optimize computer programs such as profiling, partitioning and load balancing. Common applications of parallel programming are discussed.

**Course Objectives:**

Upon successful completion of this course, students should be able to:

* Demonstrate an understanding of the concept of a thread and multithreading.
* Demonstrate an understanding of different parallel computer architectures and parallel programming models.
* Setting up and configuring a computer system to run programs concurrently.
* Profile a sequential computer program.
* Parallelize computer code in a multiprocessor computer using OpenMP.
* Parallelize computer code in a computer cluster using MPI.
* Parallelize computer code in a multi GPU system using CUDA and OpenCL.
* Demonstrate an understanding of the different strategies to parallelize and optimize serial code.
* Evaluate the performance of parallel computer programs.

**General Education Outcomes:**

**SKILLS/Inquiry/Analysis:** Students will employ scientific reasoning and logical thinking.

* **SKILLS/Communication:** Students will communicate in diverse settings and groups, using written (both reading and writing), oral (both speaking and listening), and visual means
* **VALUES, ETHICS, RELATIONSHIPS / Professional/Personal Development:** Students will work with teams, including those of diverse composition. Build consensus. Respect and use creativity.

**Pre-requisites:**

CST3503 or CST 3513 or CST 3523 – Computer Programming

**Tentative Textbook:**

Parallel Programming for Multicore and Cluster Systems (Thomas Rauber and Gudula Runger) ISBN 978-3-642-37801-0

**Academic Integrity –** Students and all others who work with information, ideas, texts, images, music, inventions, and other intellectual property owe their audience and sources accuracy and honesty in using, crediting, and citing sources. As a community of intellectual and professional workers, the College recognizes its responsibility for providing instruction in information literacy and academic integrity, offering models of good practice, and responding vigilantly and appropriately to infractions of academic integrity. Accordingly, academic dishonesty is prohibited in The City University of New York and at New York City College of Technology and is punishable by penalties, including failing grades, suspension, and expulsion. The complete text of the College policy on Academic Integrity may be found in the catalog.

**Grading Procedure:**

 Test1 15%

 Test2 15% Final Exam 30%

 Assignments/Classwork 20%

 Projects 20%

 =====

 TOTAL 100%

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Letter Grade** | A | A- | B+ | B | B- | C+ | C | D | F |
| **Numerical Grade** | 93-100 | 90-92.9 | 87-89.9 | 83-86.9 | 80-82.9 | 77-79.9 | 70-76.9 | 60-69.9 | <=59.9 |

**Course Outline:**

|  |  |  |
| --- | --- | --- |
| Week | Topics | Reading |
| 1 | Single-processor computers and programming. Threads and multithreading | Chapter 1 |
| 2 | Parallel computer architectures. | Chapter 2.1-2.7 |
| 3 | Intro to parallel programming models. | Chapter 3.1-3.3 |
| 4 | Shared-memory architectures and parallel programming with threads. | Chapter 6.1-6.2 |
| 5 | Shared-memory architectures and parallel programming with Open Multiprocessing (OpenMP). | Chapter 6.3 |
| 6 | Review and **Test#1** |  |
| 7 | Distributed memory architectures and message-passing programming with Message Passing Interface (MPI). | Chapter 5 |
| 8 | Distributed memory architectures and message-passing programming with Message Passing Interface (MPI). (continuation) | Chapter 5 |
| 9 | GPU programming using Open Computing Language (OpenCL) and CUDA. | Chapter 7 |
| 10 | Review and **Test#2** |  |
| 11 | More on parallel algorithm models and design. | Chapter 3.5-3.9 |
| 12 | Partitioning and load balancing. | Chapter 4.1-4.4 |
| 13 | Performance analysis of parallel programs. | Chapter 4.5-4.6 |
| 14 | Applications of parallel programming and computing. | Chapter 8 and other assigned readings |
| 15 | Project presentation, Review and **Final** |  |

**Assessment criteria:**

|  |  |
| --- | --- |
| **For the successful completion of this course a student should be able to:** | **Evaluation methods and criteria** |
| Demonstrate an understanding of different parallel computer architectures and parallel programming models. | Students will demonstrate a conceptual understanding of different parallel computer architectures and parallel programming models in assignments and tests. |
| Setting up and configuring a computer system to run programs concurrently. | Students will set up and configure a computer system that can run programs in parallel in laboratory assignments. |
| Parallelize computer code in a multiprocessor computer using OpenMP. | Students will write parallel programs in multiprocessor computer using OpenMP in programming tests, assignments and projects. |
| Parallelize computer code in a computer cluster using MPI. | Students will write parallel programs in a computer cluster that uses MPI in tests, programming assignments and projects. |
| Parallelize computer code in a multi GPU system using CUDA.  | Students will write parallel programs in a multi GPU system using CUDA in tests, programming assignments and projects. |
| Demonstrate an understanding of the different strategies to parallelize and optimize serial code. | Students will demonstrate a conceptual understanding and the practical application of strategies to parallelize code in programming assignments and tests. |
| Evaluate the performance of parallel computer programs. | Students will demonstrate their ability to evaluate the performance of parallel computer programs in programming assignments and tests. |

**General Education Outcomes and Assessment:**

|  |  |
| --- | --- |
| **Learning Outcomes** | **Assessment Method** |
| **SKILLS/Inquiry/Analysis** Students will employ scientific reasoning and logical thinking. | Students will describe computer problems, identify inputs, processes and desired outcomes as well as performance bottlenecks in laboratory assignments, class work and tests.Students will solve computing problems using parallel programming techniques and evaluate the code’s parallel performance in laboratory assignments, class work and tests. |
| **SKILLS/Communication**Students will communicate in diverse settings and groups, using written (both reading and writing), oral (both speaking and listening), and visual means | Final Project report in which students will write a lab report describing the code parallelization techniques used as well as a performance evaluation of the code.Oral presentation of the project |
| **VALUES, ETHICS, RELATIONSHIPS / Professional/Personal Development** Students will work with teams, including those of diverse composition. Build consensus. Respect and use creativity. | Project in which students will work in groups, build consensus and respect and use creativity |

**Other References:**

* [Principles of Parallel Programming](http://www.springer.com/computer/swe/book/978-3-642-04817-3) (Calvin Lin and Lawrence Snyder)
* [Parallel Programming](http://www.amazon.com/Parallel-Programming-Techniques-Applications-Workstations/dp/0131405632) (Wilkinson and Allen) See also [the authors' website for the book](http://www.cs.uncc.edu/~abw/parallel/par_prog/index.htm).
* [The Sourcebook of Parallel Computing](http://www.amazon.com/Sourcebook-Parallel-Computing-Kaufmann-Architecture/dp/1558608710/) (Dongarra et. al.)
* [Introduction to Parallel Computing](http://www.amazon.com/Introduction-Parallel-Computing-Ananth-Grama/dp/0201648652/) (Grama, Karypis, Kumar, and Gupta)
* [Parallel Scientific Computing in C++](http://www.amazon.com/Parallel-Scientific-Computing-MPI-Implementation/dp/0521520800/) (Karniadakis and Kirby)
* [Scientific Parallel Computing](http://amrit.ittc.ku.edu/spc/) (Scott, Clark, and Bagheri)
* [Parallel Programming in C with MPI and OpenMP](http://www.amazon.com/Parallel-Programming-C-MPI-OpenMP/dp/0072822562) (Quinn)
* [Designing and Building Parallel Programs](http://www-unix.mcs.anl.gov/dbpp/) (Foster)
* [MPI: The Complete Reference](http://www.netlib.org/utk/papers/mpi-book/mpi-book.html)
* [Using OpenMP: Portable Shared Memory Parallel Programming](http://www.amazon.com/Using-OpenMP-Programming-Engineering-Computation/dp/0262533022) (Chapman, Jost, and van der Pas)
* Multicore and GPU Programming (Gerassimos Barlas)

****

New York City College of Technology, CUNY

NEW COURSE PROPOSAL FORM

This form is used for all new course proposals. Attach this to the [Curriculum Modification Proposal Form](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/2013-10-10-Curriculum_Modification_Proposal_Form.docx) and submit as one package as per instructions. Use one New Course Proposal Form for each new course.

|  |  |
| --- | --- |
| **Course Title** | **Data Structures** |
| **Proposal Date** | 7/20/17 |
| **Proposer’s Name**  | David Bellehsen |
| **Course Number** | **CST 3650** |
| **Course Credits, Hours** | 3 credits, 4 -2 lecture hours and 2 lab hours |
| **Course Pre / Co-Requisites** | CST3503 or CST 3513 |
| **Catalog Course Description** | Introduces structure of data in computer memory, including arrays, linked lists, stacks, trees, hash tables, graphs.  Discuss algorithms to manipulate data in these structures in various ways, such as searching for a data item and sorting a set of data elements.  Covers algorithms such as how to organize data elements, how to delete, insert, edit and search for a data element in a specific data structure, how to sort a set of data elements and the differences between different data structures and when to use the right ones to solve problems. |
| **Brief Rationale**Provide a concise summary of why this course is important to the department, school or college. | As computers have become more powerful, the problems they must solve have become larger and more complex, requiring development of more intricate programs. Students will acquire good programming and algorithm analysis skills simultaneously so that they can develop such programs with maximum amount of efficiency. Students will be introduced to practical problems to solve using data structures such as linked-lists and arrays to implement a course registration tool, storing and sorting courses and students’ information data, modeling network data using a graph data structure, and applying sorting and graph algorithms to analyze network data.Students will understand the importance of the emerging field of data science and its intersection with data structures in computer science. |
| **CUNY – Course Equivalencies**Provide information about equivalent courses within CUNY, if any. | It is very common required course in CS programs Some similar courses in other CUNY colleges are listed:* CS313 – Data Structures (Queens College)
* CISC3130/CIS22 – Data Structures(Brooklyn College)
* CSC326 – Data Structures (College of Staten Island)
* CSC 21200 – Data Structures(City College)
 |
| **Intent to Submit as Common Core**If this course is intended to fulfill one of the requirements in the common core, then indicate which area. | No |
| **For Interdisciplinary Courses:*** Date submitted to ID Committee for review
* Date ID recommendation received

- Will all sections be offered as ID? Y/N | No |
|  |
|  |
| **Intent to Submit as a Writing Intensive Course** | No |

Please include all appropriate documentation as indicated in the NEW COURSE PROPOSAL Combine all information into a single document that is included in the Curriculum Modification Form.

**NEW COURSE PROPOSAL CHECK LIST**

Use this checklist to ensure that all required documentation has been included. You may wish to use this checklist as a table of contents within the new course proposal.

|  |  |
| --- | --- |
| **Completed NEW COURSE PROPOSAL FORM** |  |
| * Title, Number, Credits, Hours, Catalog course description
 | x |
| * Brief Rationale
 | x |
| * CUNY – Course Equivalencies
 | x |
| Completed [Library Resources and Information Literacy Form](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/curriculum_modification_library_form.doc) |  |
| **Course Outline** Include within the outline the following. |  |
| Hours and Credits for Lecture and LabsIf hours exceed mandated Carnegie Hours, then rationale for this | x |
| Prerequisites/Co- requisites | x |
| Detailed Course Description | x |
| Course Specific Learning Outcome and Assessment Tables* Discipline Specific
* General Education Specific Learning Outcome and Assessment Tables
 | x |
| Example Weekly Course outline | x |
| Grade Policy and Procedure | x |
| Recommended Instructional Materials (Textbooks, lab supplies, etc) | x |
| Library resources and bibliography | x |
| **Course Need Assessment.** Describe the need for this course. Include in your statement the following information. |  |
| Target Students who will take this course. Which programs or departments, and how many anticipated?Documentation of student views (if applicable, e.g. non-required elective). | x |
| Projected headcounts (fall/spring and day/evening) for each new or modified course. | x |
| If additional physical resources are required (new space, modifications, equipment), description of these requirements. If applicable, Memo or email from the VP for Finance and Administration with written comments regarding additional and/or new facilities, renovations or construction. | x |
| Where does this course overlap with other courses, both within and outside of the department? | x |
| Does the Department currently have full time faculty qualified to teach this course? If not, then what plans are there to cover this? | x |
| If needs assessment states that this course is required by an accrediting body, then provide documentation indicating that need. | NA |
| **Course Design**Describe how this course is designed.  |  |
| Course Context (e.g. required, elective, capstone) | x |
| Course Structure: how the course will be offered (e.g. lecture, seminar, tutorial, fieldtrip)? | x |
| Anticipated pedagogical strategies and instructional design (e.g. Group Work, Case Study, Team Project, Lecture) | x |
| How does this course support Programmatic Learning Outcomes? | x |
| Is this course designed to be partially or fully online? If so, describe how this benefits students and/or program. | x |
| **Additional Forms for Specific Course Categories** | NA |
|  [Interdisciplinary Form](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/Application-for-Interdisciplinary-Course-Designation.docx) (if applicable) |  |
|  Interdisciplinary Committee Recommendation (if applicable and if received)\* \*Recommendation must be received before consideration by full Curriculum Committee |  |
| [Common Core (Liberal Arts) Intent to Submit](http://openlab.citytech.cuny.edu/collegecouncil/files/2014/08/CommonCoreCourseSubmissionForm_4.2.12.doc) (if applicable) |  |
| Writing Intensive Form if course is intended to be a WIC (under development)  |  |
| If course originated as an experimental course, then results of evaluation plan as developed with director of assessment. |  |
| **(Additional materials for** [**Curricular Experiments**](http://www.300jaystreet.com/college-council/curriculum_proposals/curricular-experiments)**)** | NA |
| Plan and process for evaluation developed in consultation with the director of assessment. (Contact Director of Assessment for more information). |  |
| Established Timeline for Curricular Experiment |  |

**Course Need Assessment:**

This course will be an elective for Computer Systems Technology students. Students in Applied Mathematics and Applied Computational Physics may also benefit from this course.

One section of the course will be offered in fall and spring semesters. Expected student count is 24 students per semester.

**No additional space is necessary.**

No overlap with existing Computer Systems Technology courses.

The department has faculty members who are capable of teaching this course.

**Course Design:**

This course will be an elective for Computer Systems Technology students. Students in Applied Mathematics and Applied Computational Physics may also benefit from this course.

The course will be a combination of theory and hands-on (2 hours lecture; 2 hours lab), including the writing of computer programs to understand basic concepts and a larger group project.

**LIBRARY RESOURCES & INFORMATION LITERACY: MAJOR CURRICULUM MODIFICATION**

Please complete for **all** major curriculum modifications. This information will assist the library in planning for new courses/programs.

Consult with your library faculty subject specialist (<http://cityte.ch/dir>) **3 weeks before the proposal deadline**.

**Course proposer:** please complete boxes 1-4. **Library faculty subject specialist:** please complete box 5.

|  |  |  |
| --- | --- | --- |
| **1** | **Title of proposal****CST 3650 – Data Structures** | **Department/Program**Computer Systems TechnologyB.Tech. in Computer Systems |
|  | **Proposed by** (include email & phone)David BellehsenPhone: (718) 260-5151Email: dbellehsen@citytech.cuny.edu | **Expected date course(s) will be offered** Fall 2018**# of students 24** |

|  |  |
| --- | --- |
| **2** | **The library cannot purchase reserve textbooks for every course at the college, nor copies for all students. Consult our website (**[**http://cityte.ch/curriculum**](http://cityte.ch/curriculum)**) for articles and ebooks for your courses, or our open educational resources (OER) guide (**[**http://cityte.ch/oer**](http://cityte.ch/oer)**). Have you considered using a freely-available OER or an open textbook in this course?**Yes.  |

|  |  |
| --- | --- |
| **3** | **Beyond the required course materials, are City Tech library resources sufficient for course assignments? If additional resources are needed, please provide format details (e.g. ebook, journal, DVD, etc.), full citation (author, title, publisher, edition, date), price, and product link.**No additional resources are required beyond the required course materials.  |

|  |  |
| --- | --- |
| **4** | **Library faculty focus on strengthening students' information literacy skills in finding, critically evaluating, and ethically using information. We collaborate on developing assignments and customized instruction and research guides. When this course is offered, how do you plan to consult with the library faculty subject specialist for your area? Please elaborate.**We may consult with the library faculty specialist on an ad hoc basis, but we do not anticipate a semester long need for library faculty consultation. |

|  |  |
| --- | --- |
| **5** | **Library Faculty Subject Specialist Prof Junior Tidal\_\_\_\_\_****Comments and Recommendations**After reviewing this course proposal and surveying the library’s holdings, I would recommend more recently published monographs related to Java and C++ be added to the collection, in addition to the required texts and course bibliography. I also recommend acquiring other resources that are specific to algorithms, data containers, and sorting for these programming languages. I would also suggest that integrated development environment (IDE) and compiling software be added to library workstations to support students’ programming work. **Date 09.1217** |

**NEW YORK CITY COLLEGE OF TECHNOLOGY/CUNY**

**Computer Systems Technology Department**

**COURSE: CST3650 – Data Structures**

**( 2 class hours, 2 lab hours, 3 credits )**

##

**Course Description:**

A data structure is an arrangement of data in a computer’s memory. In this course you will learn ways to structure and arrange data in computer’s memory that includesarray, linked lists, stacks, trees, hash tables, graphs*,* among others.  Algorithms manipulate that data in these structures in various ways, such as searching for a data item and sorting a set of data elements.  In this class you will learn several concepts including how to organize data elements, how to delete, insert, edit and search for a data element in a specific data structure and how to sort a set of data elements. You will also learn the differences between different data structures and when to use to use the right ones to solve problems.

**Course Objectives:**

Upon completion of this course, student will be able to:

1. Demonstrate some understanding of object-oriented programming and be able to program with classes.
2. Demonstrate understanding of the abstract properties of various data structures such as stacks, queues, lists, and trees and be able to use these structures effectively in application programs.
3. Implement various data structures in more than one manner, compare the different implementations and explain the advantages and disadvantages of the different implementations.
4. Demonstrate understanding of and be able to program various sorting algorithms, and be able to compare the efficiency of these algorithms in terms of both time and space.
5. Trace and code recursive functions.
6. Demonstrate understanding of and be able to represent graphs, explore graphs as ADT, examine and implement various graph traversal algorithms, the shortest path algorithm, and the minimal spanning tree algorithm.

General Education Outcomes:

* **SKILLS/Inquiry/Analysis:** Students will employ scientific reasoning and logical thinking.
* **SKILLS/Communication:** Students will communicate in diverse settings and groups, using written (both reading and writing), oral (both speaking and listening), and visual means
* **INTEGRATION/Integrate learning:** Students will resolve difficult issues creatively by employing multiple systems and tools.
* **VALUES, ETHICS, RELATIONSHIPS / Professional/Personal Development:** Students will work with teams, including those of diverse composition. Build consensus. Respect and use creativity.

**Prerequisites:**

 CST3503 or CST3513

**Required Textbook:**

Data Structures and Algorithms in C++ by Goodrich, Tamassia, and Mount,

John Wiley&Sons 2004.

**Academic Integrity Policy** - Students and all others who work with information, ideas, texts, images, music, inventions, and other intellectual property owe their audience and sources accuracy and honesty in using, crediting, and citing sources. As a community of intellectual and professional workers, the College recognizes its responsibility for providing instruction in information literacy and academic integrity, offering models of good practice, and responding vigilantly and appropriately to infractions of academic integrity. Accordingly, academic dishonesty is prohibited in The City University of New York and at New York City College of Technology and is punishable by penalties, including failing grades, suspension, and expulsion. The complete text of the College policy on Academic Integrity may be found in the catalog.

**Suggestions:**

 You are encouraged to solve all the homework questions and programs on your own, but are permitted to brainstorm difficult problems in small groups, as long as each of you writes the solutions individually and honestly acknowledging the cooperation. Needless to say, if you work with others but never come up with the solution on your own, you may do OK in the homework component of your grade, but you will not do well on exams, so be careful. The use of internet, course bibles, outsiders, and other clearly “cheating” resources is strictly prohibited. Students found cheating will fail the course. If you miss a class, you are responsible for the material. Have the phone numbers of several other students to find out what material you missed

**Grading Procedure**:

Final Examination 30%

Projects 20%

Tests 30%

Assignments/Participation in class 20%

 \_\_\_\_\_\_

Total 100%

|  |  |  |
| --- | --- | --- |
|  | Topics | Reading |
| Week 1,2 | Review of C++:Object-Oriented Design | Chapter 1,2 |
| Week 3 | Analysis Tools | Chapter 3 |
| Week 4 | Review and Test1  |  |
| Week 5,6 | Stacks, Queues and Recursion | Chapter 4 |
| Week 7 | Lists | Chapter 5 |
| Week 8,9 | Trees, Search TreesReview Test2 | Chapter 6, 9 |
| Week 10 | Heaps, Hash Tables | Chapter 7,8 |
| Week 11 | Sorting, Selection | Chapter 10 |
| Week 12,13 | Graphs | Chapter 12 |
| Week 14 | Review, oral presentation of the project |  |
| Week 15 | Review and Final |  |

 **Course Outline**

**Assessment criteria**

|  |  |
| --- | --- |
| **For the successful completion of this course a student should be able to:** | **Evaluation methods and criteria** |
| 1.Understand how to use arrays, Structures | 1. Students will complete assignments writing programs that use:
* Arrays-one-dimensional and two-dimensional
* Records or Structures
* Classes, Inheritance, Polymorphism, Overloading, etc..
 |
| 2. Demonstrate an understanding of the efficiency of algorithm in both time and space. | 2. Students will demonstrate their ability to evaluate algorithms in both time and space, in assignments and project or tests. |
| 3.Demonstrate an understanding of stacks , Queues | 3. Students will write programs to search insert and delete an element from stacks and queues. Apply stacks to evaluate postfix and prefix expressions in assignments, and project or tests. |
| 4. Demonstrate an understanding of Recursion  | 4. Students will demonstrate their ability to implement recursion principles as well as showing the relationship between stacks and recursion by coding and tracing recursive functions in assignments and project. |
| 5. Demonstrate an understanding of Linked Lists | 5. Students will complete project assignment to search, insert and delete an element from a single and double linked list. Programming assignment and project will be written to implement stacks and queues as linked list.  |
| 6. Demonstrate an understanding of the use of tree as data structure, AVL trees, Multi-way search trees | 6. Students will submit programming assignment and project implementing different types of traversals.  |
| 7. Demonstrate an understanding of the various sorting and searching algorithms | 7. Students will demonstrate their ability to program various sorting and searching algorithms and compare them, in assignment and project |
| 8.Demonstrate an understanding of Graphs a data structure | 8. Students will demonstrate their ability to represent graphs, explore graphs and implement various graph traversals in assignments and project. |

**General Education Outcomes and Assessment:**

|  |  |
| --- | --- |
| **Learning Outcomes** | **Assessment Method** |
| **SKILLS/Inquiry/Analysis** Students will employ scientific reasoning and logical thinking. | Students will solve problems, using the correct algorithms in class work and tests.Students will identify coding paradigms in class work and tests |
| **SKILLS/Communication**Students will communicate in diverse settings and groups, using written (both reading and writing), oral (both speaking and listening), and visual means | In final project, students will describe the programming project and how it was implemented using an object-oriented programming Language, and the right data structureStudent will demonstrate oral presentation skill through Power Point presentations of the final programming project |
| **INTEGRATION/Integrate learning:** Students will resolve difficult issues creatively by employing multiple systems and tools. | Programming assignments where student will write codes to solve problems independently. |
| **VALUES, ETHICS, RELATIONSHIPS / Professional/Personal Development** Students will work with teams, including those of diverse composition. Build consensus. Respect and use creativity. | students will work in groups, build consensus and respect and use creativity in the final project |

**Bibliography**

1. Basic Concepts in Data Structures, By Shmuel Tomi Klein, ISBN 978-1-316-61384-9, Cambridge University Press.
2. Data Structures Using C++, By D.S. Malik, ISBN 0-619-15907-3, Thomson, Course Technology
3. Advanced Data Structures, By Peter Brass, ISBN 978-0-521-88037-4, Cambridge University Press
4. Inroduction to Java Programming and Data Structures, 11th Ed, By Y. Daniel Liang, ISBN-13: 978-0-13-467094-2, Pearson
5. Data Structures and Algorithm Analysis in JAVA, 3rd Ed, By Mark Allen Weiss, ISBN10 0-13-257627-9, Pearson
6. Object-Oriented Data Structures Using Java, 4th ed, By Nell Dale, Daniel T. Joyce, Chip Weems ISBN-13: 9781449613549, Jones & Bartlett Learning
7. Data Structures and Algorithms in Java, By Adam Drozdek, ISBN 0-534-37668-1, Brooks/Cole Thomson Learning
8. Data Structures Using C and C++, 2nd Ed, By Langsam, Augenstein, and Tenenbaum,

ISBN 0-13-036997-7, Prentice Hall

1. C++ An Introduction to Data Structures, 2nd Ed, By Larry Nyhoff, ISBN 0-02-388725-7, Prentice Hall
2. Data Structures with C++ Using STL, 2nd Ed, By Ford and Topp, ISBN 0-13-085850-1, Prentice Hall



