KINGSBOROUGH COMMUNITY COLLEGE
The City University of New York

CURRICULUM TRANSMITTAL COVER PAGE

Department: Math and Computer Science **Date:** 01/14/2019

Title Of Course/Degree/Concentration/Certificate: Data Structures (CS 3700)

Change(s) Initiated: (Please check)

- □ Closing of Degree
- □ Closing of Certificate
- □ New Certificate Proposal
- □ New Degree Proposal
- □ New Course
- □ New 82 Course (Pilot Course)
- □ Deletion of Course(s)
- □ Change in Degree or Certificate
- □ Change in Degree: Adding Concentration
- □ Change in Degree: Deleting Concentration
- □ Change in Prerequisite, Corequisite, and/or Pre/Co-requisite
- □ Change in Course Designation
- □ Change in Course Description
- ■ Change in Course Title, Number, Credits and/or Hours
- □ Change in Academic Policy
- □ Pathways Submission:
  - □ Life and Physical Science
  - □ Math and Quantitative Reasoning
  - □ A. World Cultures and Global Issues
  - □ B. U.S. Experience in its Diversity
  - □ C. Creative Expression
  - □ D. Individual and Society
  - □ E. Scientific World
- □ Change in Program Learning Outcomes
- □ Other (please describe):____________

PLEASE ATTACH MATERIAL TO ILLUSTRATE AND EXPLAIN ALL CHANGES

DEPARTMENTAL ACTION

Action by Department and/or Departmental Committee, if required:

Date Approved:__________ **Signature, Committee Chairperson:**____________

If submitted Curriculum Action affects another Department, signature of the affected Department(s) is required:

Date Approved:__________ **Signature, Department Chairperson:**____________

Date Approved:__________ **Signature, Department Chairperson:**____________

I have reviewed the attached material/proposal

Signature, Department Chairperson: 1/14/2019

Kingsborough Community College
The City University of New York

Modifications in Credits/Hours for an Existing Course Form

1. Course Number and Title:
   Mathematics and Computer Science
   CS 3700 - Data Structures

2. This Course is **currently** listed as:

   __4__ Credits __4__ Hours (include break-down of lecture, lab, or gym)
   4 Lecture Hours

3. **Proposed** Change in Credits/Hours (Please check **ONE** appropriate box below based on credits):

   It is recommended that you refer to the "College Credits Assigned for Instructional Hours" PDF at **http://kingsborough.edu/aa/Pages/forms.aspx**

   Hours are hours per week in a typical 12-week semester

1-credit:
□ 1 hour lecture
□ 2 hours lab/field/gym

2-credits:
□ 2 hours lecture
□ 1 hour lecture, 2 hours lab/field
□ 4 hours lab/field

3-credits:
□ 3 hours lecture
☒ 2 hours lecture, 2 hours lab/field
□ 1 hour lecture, 4 hours lab/field
□ 6 hours lab/field

4-credits:
□ 4 hours lecture
□ 3 hours lecture, 2 hours lab/field
□ 2 hours lecture, 4 hours lab/field
□ 1 hour lecture, 6 hours lab/field
□ 8 hours lab/field

More than 4-credits: □ Number of credits: ____ (explain mix lecture/lab below)

___ Lecture ____Lab

Explanation:____

4. Rationale/Justification for the change in credits/hours for this course:

   The change in number of credits reflects curricular adjustments to allow for 2 lab hours and 2 lecture hours, as reflected in the course syllabus.

5. Include the **<u>Current</u>** Syllabus/Topical Course Outline and the **<u>Proposed</u>** Syllabus/Topical Course Outline for the course. **Highlight** areas that have been modified and serve as the justification for the proposed change in credits/hours for the course.

   See attachments.

KINGSBOROUGH
COMMUNITY COLLEGE
★ DREAMS BEGIN HERE ★

TO: Spring 2019 Curriculum Committee

FROM: Department of Mathematics & Computer Science

DATE: 01/14/2019

RE: Change in Number of Course Credits for Data Structures (CS 3700)

The Department of Mathematics & Computer Science is proposing a change in Number of Course Credits for Data Structures (CS 3700):

FROM:

4 credits, 4 hrs.

TO:

3 credits, 4 hrs. (2 lecture hrs., 2 hr. lab)

Rationale for Change: The change in number of credits reflects curricular adjustments to allow for 2 lab hours and 2 lecture hours, as reflected in the course syllabus.

CURRENT

Kingsborough Community College
of the City University of New York

Department of Mathematics & Computer Science

1. **Department, Course Number and Title**
   Department of Mathematics & Computer Science, CS 37 - Program Design and Analysis

2. **Distribution Requirements for Groups I-V**
   This course satisfies the Group V requirement.

3. **Demonstration of Course Transferability**
   <go to TIPS web site> http://student.cuny.edu

4. **Bulletin Description of Course**
   An introduction to data structures using C++. Topics: abstract data types and the use of aggregate data structures with practical implementation of static as well as dynamic data structures including: stacks, queues, linked lists, and trees. Search and sort algorithms are investigated and applied. These elements of computer science are a critical area of study which prepare computer scientists for real-world programming and software engineering tasks. This course teaches the fundamentals of organizing and manipulating data efficiently using clean conceptual models.

5. **Number of Weekly Class Hours**
   4

6. **Number of Credits**
   4

7. **Prerequisite**
   CS 13

8. **Justification for Course and Expected Enrollment**
   This course will be offered as an elective for computer science majors. It is in concurrence with the ACM (Association for Computing Machinery) guidelines for programming courses using Data Structures.

9. **Course Withdrawals**
   none

10. **CPI Requirements**
    N/A

11. **Field Work, Internship or Independent Study**
    N/A

12. **Textbook**

ADTs, Data Structures, and Problem Solving with C++, 2nd Ed., Nyhoff, Larry R., Prentice Hall, 2004

13. **Required Course for Majors**

Not required.

14. **Specify If Course Is Open to Only Selected Students**

All students who have passed CS 13.

15. **What Students Will Know and Be Able To Do Upon Completion of Course**

The students will develop an understanding of data structures; the students will learn to implement various structure representations in their code; the student will be able to write and execute object oriented C++ programs.

16. **Method of Teaching**

Lecture, discussion, lab sessions.

17. **Assignments to Students**

Readings and exercises from the texts; computer programs.

18. **Method of Evaluating Learning**

Examinations and assignments.

19. **Topical Course Outline**

1. Abstract Data Types
– struct and class implementations of ADT's and aggregate data structures.
– implementation examples with struct and class passed as function parameters

2. Programming with Dynamic Data Structures
– pros and cons of dynamic vs static data structures,
introduction to memory management issues with DDS

3. the Linked List - The list interface, using list objects, iterators, applications singly linked, doubly linked, circularly linked lists and their implementations

4. Stacks – LIFO, the stack interface, using stack objects, application of stacks, stack applied to expression handling: infix/postfix/prefix representations

5. Queues – FIFO, the queue interface, using queue objects, applications of queues

6. Trees - tree terminology, decision trees and transition diagrams, tree traversal algorithms, implementation of the tree class

7. Searching - Binary search trees, Depth First vs. Breadth First searchs, Linear Search vs. Binary Search.

8. Sorting – efficiency of sorting algorithms, intro to BigO noation [n^2 vs. n logn complexity] bubble/selection/insertion/merge/heap and quicksort.

20. Bibliography

Data Structures using C and C++, Langsam, Y., Augenstein, M., and Tennenbaum, A., Prentice Hall, 1996.

Gaddis, T., *Standard Version of Starting Out with C++,* 4th Edition, Scott Jones Publishers

C++ Plus Data Structures 6th Edition (2016), Dale, N., Weems, C., Richards, T.

Data Structures &Algorithms C++ (Prentice-Hall Object-Oriented Series), Rowe, G.

Updated: Nov. 2005
G. Bassen

Update: June 2016
P. Listowsky in consultation with G. Bassen

KINGSBOROUGH COMMUNITY COLLEGE
THE CITY UNIVERSITY OF NEW YORK

Proposed

COURSE SYLLABUS: CS 3700

1. **DEPARTMENT, COURSE NUMBER, AND TITLE:**
   Mathematics and Computer Science
   CS 3700 - Data Structures

2. **DOES THIS COURSE MEET A GENERAL EDUCATION/CUNY CORE CATEGORY?**
   □ **Life and Physical Science**
   □ **Math and Quantitative Reasoning**
   □ **A. World Cultures and Global Issues**
   □ **B. U.S. Experience in its Diversity**
   □ **C. Creative Expression**
   □ **D. Individual and Society**
   □ **E. Scientific World**

   IF YES, COMPLETE AND SUBMIT WITH THIS PROPOSAL A CUNY COMMON CORE SUBMISSION FORM.

3. **DESCRIBE HOW THIS COURSE TRANSFERS (REQUIRED FOR A.S. DEGREE COURSE). IF A.A.S. DEGREE COURSE AND DOES NOT TRANSFER, JUSTIFY ROLE OF COURSE, E.G. DESCRIBE OTHER LEARNING OBJECTIVES MET:**

   Course CS 3700 (Data Structures) is equivalent to:
   CSCI 373 Advanced Data Structures - John Jay College of Criminal Justice
   CSC 326 Data Structures - College of Staten Island
   CSI 33 Data Structures - Bronx Community College
   CSC 331 Data Structures - Borough of Manhattan Community College

4. **BULLETIN DESCRIPTION OF COURSE:**

   An introduction to data structures using C++. Topics: abstract data types and the use of aggregate data structures with practical implementation of static as well as dynamic data structures including: stacks, queues, linked lists, and trees. Search and sort algorithms are investigated and applied. These elements of computer science are a critical area of study which prepare computer scientists for real-world programming and software engineering tasks. This course teaches the fundamentals of organizing and manipulating data efficiently using clean conceptual models.

5. **CREDITS AND HOURS* (PLEASE CHECK ONE APPROPRIATE BOX BELOW BASED ON CREDITS):**

1-credit:	□ 1 hour lecture □ 2 hours lab/field/gym
2-credits:	□ 2 hours lecture □ 1 hour lecture, 2 hours lab/field

	□ 4 hours lab/field

3-credits:	□ 3 hours lecture ☒ 2 hours lecture, 2 hours lab/field □ 1 hour lecture, 4 hours lab/field □ 6 hours lab/field

4-credits:	□ 4 hours lecture □ 3 hours lecture, 2 hours lab/field □ 2 hours lecture, 4 hours lab/field □ 1 hour lecture, 6 hours lab/field □ 8 hours lab/field

More than 4-credits: □ Number of credits: ____ (explain mix lecture/lab below)

___ Lecture ____Lab

Explanation:____

***Hours are hours per week in a typical 12-week semester**

6. **NUMBER OF EQUATED CREDITS IN ITEM #5:________**

7. **COURSE PREREQUISITES AND COREQUISITES (IF NONE PLEASE INDICATE FOR EACH)**
   A. PREREQUISITE(S): CS 13A0
   B. COREQUISITE(S): N/A
   C. PRE/COREQUISITE(S): N/A

8. **BRIEF RATIONALE TO JUSTIFY PROPOSED COURSE TO INCLUDE:**
   A. ENROLLMENT SUMMARY IF PREVIOUSLY OFFERED AS AN 82 (INCLUDE COMPLETE 4-DIGIT 82 COURSE NUMBER)
   B. PROJECTED ENROLLMENT: 28 students
   C. <u>SUGGESTED</u> CLASS LIMITS: 28 students
   D. FREQUENCY COURSE IS LIKELY TO BE OFFERED: Fall and Spring semesters.
   E. ROLE OF COURSE IN DEPARTMENT'S CURRICULUM AND COLLEGE'S MISSION: Key in adding value to Kingsborough Community College's A.S. Computer Science degree; introduces vital topics in field, and methods & practices.

9. **LIST COURSE(S), IF ANY, TO BE WITHDRAWN WHEN COURSE IS ADOPTED (NOTE THIS IS NOT THE SAME AS DELETING A COURSE):** N/A

10. **IF COURSE IS AN INTERNSHIP, INDEPENDENT STUDY, OR THE LIKE, PROVIDE AN EXPLANATION AS TO HOW THE STUDENT WILL EARN THE CREDITS AWARDED. THE CREDITS AWARDED SHOULD BE CONSISTENT WITH STUDENT EFFORTS REQUIRED IN A TRADITIONAL CLASSROOM SETTING:** N/A

11. **PROPOSED TEXT BOOK(S) AND/OR OTHER REQUIRED INSTRUCTIONAL MATERIAL(S):**
    Data Abstraction and Problem Solving with C++: Walls And Mirrors
    Frank M. Carrano, Timothy M. Henry, Pearson, 2017. ISBN-10: 0-13-446397-8

12. **REQUIRED COURSE FOR MAJOR OR AREA OF CONCENTRATION?**

Required for the A.S. Computer Science degree. It is in concurrence with the ACM (Association for Computing Machinery) guidelines for programming courses using Data Structures.

13. If Open Only to Selected Students Specify Population: N/A

14. Explain what students will know and be able to do upon completion of course:
The students will develop an understanding of data structures; the students will learn to implement various structure representations in their code; the student will be able to write and execute high quality object oriented C++ programs of substance and practical use.

15. Methods of Teaching –e.g. lectures, laboratories, and other assignments for students, including any of the following: demonstrations, group work, Website or E-mail interactions and/or assignments, practice in application of skills, etc.:
Lecture, discussion, lab sessions. Computer Programming assignments.

16. Assignments to Students:
Computer programs and various assignments. Lecture, discussion and laboratory exercises are designed specifically to prepare students to create complex meaningful programs. As the semester progresses, assignments will build on previous lessons.

17. describe method of evaluating learning specified in #15 - Include percentage breakdown for grading.
Class exams, final exam, lab assignments, programming projects and exercises.
Course Grade:
Exam 1 (25%), Exam 2 (25%), Programming Assignments & Attendance (15%), and Final Exam (35%).

18. Topical Course Outline for the 12 week semester (which should be specific regarding topics covered, learning activities, and assignments):

Week 1. Abstract Data Types: (lectures plus 2 hours computer lab programming)
– struct and class implementations of ADT's and aggregate data structures.
– implementation examples with struct and class passed as function parameters

Week 2. Programming with Dynamic Data Structures (lectures plus 2 hours computer lab programming) pros and cons of dynamic vs static data structures, introduction to memory management issues with DDS

Week 3. the Linked List - The list interface, using list objects, iterators, applications singly linked, doubly linked, circularly linked lists and their implementations (lectures plus 2 hours computer lab programming)

Week 4. Linked List implementations (lectures plus 2 hours computer lab programming)

Week 5. Stacks – LIFO, the stack interface, using stack objects, application of stacks, stack applied to expression handling: infix/postfix/prefix representations (lectures plus 2 hours computer lab programming)

Week 6. Stack Implementations (lectures plus 2 hours computer lab programming)

Week 7. Queues – FIFO, the queue interface, using queue objects, applications of queues

(lectures plus 2 hours computer lab programming)

Week 8. Queue implementations (lectures plus 2 hours computer lab programming)

Week 9. Trees - tree terminology, decision trees and transition diagrams, tree traversal algorithms, implementation of the tree class (lectures plus 2 hours computer lab programming)

Week 10. Tree implementations (lectures plus 2 hours computer lab programming)

Week 11. Searching - Binary search trees, Depth First vs. Breadth First searchs, Linear Search vs. Binary Search. (lectures plus 2 hours computer lab programming)

Week 12. Sorting – efficiency of sorting algorithms, intro to BigO noation [n^2 vs. n logn complexity] bubble/selection/insertion/merge/heap and quicksort. (lectures plus 2 hours computer lab programming)

NOTE: Week 11 and 12 (Searching and Sorting) may be moved earlier in the sequence, at the discretion of instructor. Position of these critical topics should be at any point after week 2.

19. SELECTED BIBLIOGRAPHY AND SOURCE MATERIALS:

D. Malhotra and N. Malhotra, *Data Structures and Program Design Using C++*, 2019

Cormen T.H. and Leiserson C.E., *Introduction to Algorithms*, 3rd Edition (The MIT Press), 2009

Parker A., *Algorithms and Data Structures in C++* (Computer Science & Engineering Book 5), 2018

Anggoro W., *C++ Data Structures and Algorithms: Learn how to write efficient code to build scalable and robust applications in C++,* 2018

Drozdek, *Data Structures and Algorithms in C++,* 2nd Edition, 2018

Broquard V., *Beginning Data Structures in C++*, 2014

Bancila, *The Modern C++ Challenge: Become an expert programmer by solving real-world problems*, 2018

Langsam, Y., Augenstein, M., and Tennenbaum, A., *Data Structures using C and C++,* Prentice Hall, 1996.

Gaddis, T., *Standard Version of Starting Out with C++*, 4th Edition, Scott Jones Publishers

Dale, N., Weems, C., Richards, T., *C++ Plus Data Structures,* 6th Edition (2016), Rowe, G., *Data Structures &Algorithms C++,* (Prentice-Hall Object-Oriented Series),

Cormen T., Lieserson C., Rivest R., *Introduction to Algorithms, second edition*, MIT Press, Cambridge, Mass., 1993.

Guzdial M. and Ericson B., *Problem Solving with Data Structures Using Java,* Prentice Hall, Upper Saddle River, N.J., 2011.

Horowitz E. and Sahni S., *Fundamentals of Computer Algorithms,* Computer Science Press, Rockville, Maryland, 1978.

Lewis J. and Chase J, *Java Software Structures,* 3rd ed. (the shipping container book), Addison-Wesley, Reading, Mass., 2010.

Mehlhorn K. Sanders P., *Algorithms and Data Structures,* Springer, 2008.

Wirth N., *Algorithms + Data Structures = Programs,* Addison-Wesley, Reading, Mass., 1976.

Brassard G. and Bratly P, *Algorithmics: Theory and Practice,* Prentice Hall, Englewood Cliffs, New Jersey, 1988.

Revised/January 2019/Philip Listowsky