* SECTION 1 to be completed by CURRICULUM AUTHOR
* SECTION 2 to be completed by CHAIR/DIRECTOR
* SECTION 3 to be completed by DEAN
* SECTION 4 to be completed by REGISTRAR & STAFF

**SUBMIT TO AREA ADMIN. ASSISTANT AFTER SECTIONS 1-4 ARE COMPLETE TO BEGIN SIGNATURE PORTION OF THE PROCESS.**

* SECTION 5 to be completed by CURRICULUM COUNCIL CHAIR & RECORDER
* SECTION 6 to be completed by VPI

| **SECTION 1: AUTHOR** | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| New, Change or Archive? | | | Modify/Change | | | **Rationale/reasons for this request:**  1. Variable credit - background and justification attached  2. Update MCO for accuracy and relevance | | | | | | |
| Effective Quarter: | | | Fall 2024 | | |
| If proposed course will replace current, when do we archive current course? | | | N/A | | |
| **CURRENT** | | | | | | **PROPOSED** | | | | | | |
| Subject and Catalog # | | | | | | | | | | | | |
| Course Subject and Number: | | | CS 240 | | | Course Subject and Number: | | | -unchanged- | | | |
| ctcLink Course Number: | | | 192069 | | | ctcLink Course Number: | | | -unchanged- | | | |
| Short Course Title (limit 30 characters) | | | | | | | | | | | | |
| Javascript II | | | | | | -unchanged- | | | | | | |
| Long Course Title | | | | | | | | | | | | |
| -unchanged- | | | | | | -unchanged- | | | | | | |
| Course Description | | | | | | | | | | | | |
| CS 240 is the second language in a series of two courses designed to give students an extension of the current language and an introduction and implementation of data structures including queues, stacks, trees and graphs, using the current programming language. Topics include iterative and recursive implementations. | | | | | | Students learn the fundamentals of algorithm data structures within JavaScript (JS) and server-side JS. Students build template-assembled web pages, store and retrieve site information from databases, and build RestFul API endpoints. Students perform data handling of forms, usage of middleware, and user-authentication. | | | | | | |
| Course Learning Outcomes | | | | | | | | | | | | |
| * Contrast the concept of structured programming against OOP. * Design programs using the concepts of structured programming. * Develop algorithms to solve complex problems. * Use arrays, structures, stacks, and queues. * Explain and classify the different classes withing Object Oriented Programming. * Demonstrate sorting and searching methods. * Design higher order functions. * Explain ES6 semantics and syntax additions. * Use a transpiler. * Test JavaScript in the Browser. * Analyze JavaScript in NodeJS. | | | | | | * Create industry-grade JavaScript code using common design patterns * Use basic data structures such as arrays, lists, stacks, queues, and trees (optional) * Use of common sorting and searching algorithms: graphs, hash tables, heaps, linked-lists, queues, sets, stacks * Use server-side JS to create JavaScript web applications * Maintain full-stack JS application codebases using a package manager to oversee modules * Create web application servers using Javascript based routing | | | | | | |
| Course Topics | | | | | | | | | | | | |
| * OOP principles used for JavaScript coding practices. * Debugging JavaScript in the browser. * Advanced Functions: Callback functions, Bind, Curry, IIFE (Immediately Invoked Function Expression) * Asynchronous execution, timers, Prototypal Inheritance, and Closures * JavaScript Design Patterns, Object Creation Patterns, and Code Reuse Patterns * Code Minification and Compression, Loading Strategies, and Dependency Management * First-class and higher order functions in coding practices. * Functional programming techniques to your JavaScript projects. | | | | | | * Review of OOP principles * Review of FP principles * Asynchronous design patterns * Commonly use data sorting Algorithm * Graphs, Hash tables, Heaps, Linked-lists, Queues, Sets, Stacks * Basic Algorithm performance: O(n), O(log n), O(1) * Algorithm modeling in Node.js Ecosystem * Code review of arrays, lists, stacks, and queues * Node.js * Modules * Asynchronous patterns * Events * Simple HTTP Server * Express.js * Server-side rendered application with templates * API RESTful endpoints * Server routing * Middleware design patterns * Authentication Methods | | | | | | |
| Total Credits/Variable Credit | | | | | | | | | | | | |
| Minimum: | 5 | | Maximum: | | 5 | Minimum: | 1 | | Maximum: | | 5 | |
| Course Components | | | | | | | | | | | | |
| Course Components: | | | LEC | | | Course Components: | | | LEC | | | |
| Quarters Offered | | | | | | | | | | | | |
| Year 1 (list quarters): | | | -unchanged- | | | Year 1 (list quarters): | | | -unchanged- | | | |
| Year 2 (list quarters): | | | -unchanged- | | | Year 2 (list quarters): | | | -unchanged- | | | |
| Miscellaneous | | | | | | | | | | | | |
| Permit Open Entry/Exit? | | | | No | | Permit Open Entry/Exit? | | | | No | | |
| Grading Basis: | | | | Choose an item. | | Grading Basis: | | | | Graded | | |
| Instructor/dept. consent required? | | | | Choose an item. | | Instructor/dept. consent required? | | | | No Consent | | |
| Hide from Class Search? | | | | Choose an item. | | Hide from Class Search? | | | | Visible in Search | | |
| ***Approved By:*** | |  | | | | ***Date:*** | |  | | | |

| **SECTION 2: CHAIR/DIRECTOR** | | | |
| --- | --- | --- | --- |
| Enrollment Requirements & Equivalencies | | | |
| Prerequisite: |  | Pre-requisite: |  |
| Co-requisite: |  | Co-requisite: |  |
| Equivalencies: |  | Equivalencies: |  |
| Cross-Listed Courses: |  | Cross-Listed Courses: |  |
| *Notes and comments:* | | | |
| ***Approved By:*** |  | ***Date:*** |  |

| **SECTION 3: DEAN** | | | | | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CURRENT** | | | | | | | | **PROPOSED** | | | | | | | | | | |
| CONFIRM: Total Credits/Variable Credit | | | | | | | | | | | | | | | | | | |
| Minimum: |  | | Maximum: | |  | | | Minimum: |  | | Maximum: | | | |  | | | |
| Components & Credit-to-Hour Details | | | | | | | | | | | | | | | | | | |
|  | | | | CR | |  | HRS |  | | | | | CR | | |  | HRS | |
| LEC/[Theory](https://www.sbctc.edu/colleges-staff/policies-rules/policy-manual/chapter-5.aspx) (1 CR = 1 hour/week): | | | |  | | x1= |  | LEC/[Theory](https://www.sbctc.edu/colleges-staff/policies-rules/policy-manual/chapter-5.aspx) (1 CR = 1 hour/week): | | | | |  | | | x1= |  | |
| LAB/Guided Practice (1 CR = 2 hours/week): | | | |  | | x2= |  | LAB/Guided Practice (1 CR = 2 hours/week): | | | | |  | | | x2= |  | |
| CLIN/Field-Based (1 CR = 3 hours/week): | | | |  | | x3= |  | CLIN/Field-Based (1 CR = 3 hours/week): | | | | |  | | | x3= |  | |
| Total Credits & Weekly Contact Hours: | | | |  | |  |  | Total Credits & Weekly Contact Hours: | | | | |  | | |  |  | |
| Coding & Financials | | | | | | | | | | | | | | | | | | |
| [CIP](https://www.sbctc.edu/resources/documents/colleges-staff/data-services/data-warehouse/cip-2020-descriptions.pdf) Code: | | |  | | | | | [CIP](https://www.sbctc.edu/resources/documents/colleges-staff/data-services/data-warehouse/cip-2020-descriptions.pdf) Code: | | | |  | | | | | | |
| Default Section Size/Cap: | | |  | | | | | Default Section Size/Cap: | | | |  | | | | | | |
| Course Attributes: | | | Choose an item. | | | | | Course Attributes: | | | | Choose an item. | | | | | |
| Course Attributes: | | | Choose an item. | | | | | Course Attributes: | | | | Choose an item. | | | | | |
| Course Attributes: | | | Choose an item. | | | | | Course Attributes: | | | | Choose an item. | | | | | |
| Fees identified/requested: | | |  | | | | | Fees identified/requested: | | | |  | | | | | | |
| Does a budget exist for this subject? | | | Choose an item. | | | | | Does a budget exist for this subject? | | | | Choose an item. | | | | | | |
| Dean is required to present course to applicable instructional faculty (all campuses) for consideration. Date Presented: | | | | | | | | | | | | | |  | | | | |
| *Notes and comments:* | | | | | | | | | | | | | | | | | | |
| ***Approved By:*** | |  | | | | | | ***Date:*** | |  | | | | | | | | |

| **SECTION 4: REGISTRAR** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **CURRENT** | | **PROPOSED** | | | |
| Coding & Attributes | | | | | |
| Academic Group: | Choose an item. | | Academic Group: | Choose an item. | |
| Academic Org: | Choose an item. | | Academic Org: | Choose an item. | |
| Funding Source: | Choose an item. | | Funding Source: | Choose an item. | |
| Intent: | Choose an item. | | Intent: | Choose an item. | |
| Mult Enrl in Term: | Choose an item. | | Mult Enrl in Term: | Choose an item. | |
| # Completions Allowed: | Choose an item. | | # Completions Allowed: | | Choose an item. | |
| Enrollment Req Group: |  | | Enrollment Req Group: |  | |
| *Notes and comments:* | | | | | |
| ***Approved By:*** |  | | ***Date:*** |  | |

| **SECTION 5: CURRICULUM COUNCIL** | | | |
| --- | --- | --- | --- |
| ***Date Received by Council:*** |  | ***Notes and comments:*** | |
| ***Curriculum Council Reviewer*** |  |
| ***Recorded Vote:*** |  | ***Date:*** |  |
| ***Curriculum Council Chair:*** |  | ***Date Certified:*** |  |

| **SECTION 6: VICE PRESIDENT OF INSTRUCTION** | | | |
| --- | --- | --- | --- |
| ***Notes and comments:*** | | | |
| ***VP of Instruction:*** |  | ***Date:*** |  |