



ROCHESTER INSTITUTE OF TECHNOLOGY
MINOR PROGRAM PROPOSAL FORM

**B. THOMAS GOLISANO COLLEGE OF
COMPUTING AND INFORMATION SCIENCES**

Name of Certifying Academic Unit; Department of Computer Science

Name of Minor: Computer Science

Brief description of the minor to be used in university publications

Before beginning a minor in Computer Science, students must possess prerequisite knowledge that can be obtained from various programming sequences and courses in calculus and discrete mathematics. The minor itself consists of one required course and then four electives chosen by a student. Depending on a student's interests, they can select sets of electives that are more theoretically oriented, more programming oriented, targeted toward a subfield with Computer Science, such as intelligent systems, data management, or graphics, or designed to provide a broad overview of the field.

1.0 Minor Program Approvals

	Approval request date:	Approval granted date:
Academic Unit Curriculum Committee	December 5, 2011	December 5, 2011
College Curriculum Committee	January 13, 2012*	February 10, 2012
Inter-College Curriculum Committee	February 10, 2012	February 13, 2012

(*) Amended January 25, 2012; February 3, 2012

2.0 Rationale:

A minor at RIT is a related set of academic courses consisting of no fewer than 15 semester credit hours leading to a formal designation on a student's baccalaureate transcript

How is this set of academic courses related?

Some background is necessary in order to understand how and why the proposed Computer Science minor under semesters differs significantly from the current Computer Science minor under quarters.

While there are numerous factors that influenced our proposal, two factors in particular stand out. First, Computer Science had planned to update its curriculum and had already developed a proposal to do so prior to the announcement that RIT would switch to semesters. We used semester conversion to improve and complete our planned changes – the resulting B.S. degree program in Computer Science under semesters is considerably

different from the current B.S. degree program in Computer Science under quarters, especially with respect to required Computer Science courses (and to a lesser extent, with respect to required Mathematics courses). While it is not necessary to review the entire B.S. degree program under semesters, it is instructive to review the relationships among all of the required Computer Science and required Mathematics courses that comprise this program – we have reproduced a figure (that was included in our presentation of the proposed B.S. degree under semesters to the ICC) and added an explanatory table at the end of this form.

Second, in our read of the RIT minors policy approved during fall, 2011, we noted the following statement:

“In most cases, minors shall consist of a **minimum of two upper division courses** to provide reasonable breadth and depth within the minor.” (emphasis added)

We believe that the changes made to our minor resulted primarily from the combination of these two factors. In order for students to successfully complete a related set of Computer Science courses that include some upper-level courses, they need to complete not only various prerequisites, but one required course in Computer Science.

The following table illustrates differences between the proposed Computer Science minor under semesters and the current Computer Science minor under quarters.

Side-by-side comparison	
Current Computer Science Minor - Quarters	Proposed Computer Science Minor – Semesters
<ul style="list-style-type: none"> • <i>At least 20 QH (typically five courses)</i> • All courses taken in CS at RIT • <i>At least 12 QH in courses not required by a student's home department</i> • Minimum GPA of 2.0 in the minor courses • A few CS courses not approved for use in the minor • Graduate CS courses are permitted 	<ul style="list-style-type: none"> • <i>At least 15 SH (typically five courses)</i> • All courses taken in CS at RIT • <i>At least 9 SH in courses not required by a student's home department</i> • Minimum GPA of 2.0 in the minor courses • A few CS courses not approved for use in the minor • Graduate CS courses are permitted
<i>No prerequisites</i> , but students must adhere to individual course prerequisites or gain instructor permission	<i>Includes prerequisites</i> (which may be satisfied in multiple ways): <ul style="list-style-type: none"> • A two-course introductory programming sequence • A two-course calculus sequence • A course in discrete mathematics
<i>No required courses</i>	<i>One required course:</i> <ul style="list-style-type: none"> • CSCI 243 (The Mechanics of Programming)
<i>No requirement that courses be related or</i>	Students will be advised to choose

<i>those courses be upper level, although often that turned out to be the case.</i>	collections of <i>courses that tend to support a theme and that contain at least two upper-level Computer Science courses</i> (under semesters, courses numbered 300 and above should meet this requirement). In some cases, a student may be able to complete a CS minor by taking only one upper-level Computer Science course, but this will require approval from the minor advisor.
---	--

3.0 Multidisciplinary involvement:

If this is a multidisciplinary minor spanning two or more academic units, list the units and their role in offering and managing this minor.

Not applicable.

4.0 Students ineligible to pursue this minor:

The purpose of the minor is both to broaden a student's college education and deepen it in an area outside the student's major program. A minor may be related to and complement a student's major, or it may be in a completely different academic/professional area. It is the responsibility of the academic unit proposing a minor and the unit's curriculum committee to indicate any home programs for which the minor is not a broadening experience.

Please list below any home programs whose students will not be allowed to pursue this minor, provide the reasoning, and indicate if this exclusion has been discussed with the affected programs:

Computer Science majors are **prohibited** from pursuing this minor, as '[t]he purpose of the minor is both to broaden a student's college education and deepen it in an area *outside* (emphasis added) the student's major program.' No other programs are excluded.

5.0 Minor Program Structure, Sequence and Course Offering Schedule:

Describe the structure of the proposed minor and list all courses, their anticipated offering schedule, and any prerequisites.

- All minors must contain at least fifteen semester credit hours;
- Minors may be discipline-based or interdisciplinary;
- In most cases, minors shall consist of a minimum of two upper division courses (300 or above) to provide reasonable breadth and depth within the minor;

- As per New York State requirements, courses within the minor must be offered with sufficient frequency to allow students to complete the minor within the same time frame allowed for the completion of the baccalaureate degree;
- Provide a program mask showing how students will complete the minor.

Narrative of Minor Program Structure:

I)	Students must satisfy the following prerequisites (1):
a)	A two-course introductory programming sequence, such as one of the following: <ul style="list-style-type: none"> • CSCI 141 (Computer Science I) and CSCI 142 (Computer Science II) (the standard requirement) • ISTE 120 (Computational Problem Solving in the Information Domain I) and ISTE 121 (Computational Problem Solving in the Information Domain II) • ISTE 100 (Computational Problem Solving in the Network Domain I) and ISTE 101 (Computational Problem Solving in the Network Domain II) • IGME 105 (Game Software Development I) and IGME 106 (Game Software Development II) • CPET 121 (Computational Problem Solving I) and CPET 321 (Computational Problem Solving II) • Or an equivalent sequence as determined by the Computer Science minor advisor
b)	A two-course Calculus sequence, such as one of the following: <ul style="list-style-type: none"> • MATH 181 (Project-Based Calculus I) and MATH 182 (Project-Based Calculus II) (the standard requirement) • Or an equivalent sequence as determined by the Computer Science minor advisor
c)	A course in Discrete Mathematics, such as one of the following: <ul style="list-style-type: none"> • MATH 190 (Discrete Mathematics for Computing) (the standard requirement) (note, this course is one of the prerequisites for CSCI 250, CSCI 261, CSCI 262, CSCI 320, and CSCI 344; all of these courses could potentially be included by students in a Computer Science minor) • MATH 192 (Discrete Mathematics and Introduction to Proof) • MATH 131 (Discrete Mathematics) • Or an equivalent course as determined by the Computer Science minor advisor
II)	Students must complete the following required course (2): <ul style="list-style-type: none"> • CSCI 243 (The Mechanics of Programming) (note, CSCI 243 is one of the prerequisites for CSCI 250, CSCI 251, CSCI 261, and CSCI 344)
III)	Students will complete the Computer Science minor by taking four additional Computer Science courses for which they have the prerequisites. We believe, subject to specific exclusions (listed below), that Computer Science courses numbered 300-499 (upper division undergraduate courses),

500-599 (advanced upper division undergraduate courses), 600-699 (introductory graduate courses), and 700-799 (advanced graduate courses), may all be taken as part of the minor and count toward the requirement of “two upper division courses.”

We will not define all of the possible combinations of Computer Science courses that might be used for a minor, although we suggest several likely combinations below. Rather, we will seek to advise students and help them create collections of courses that are appealing and useful to their specific goals. The following example scenarios demonstrate the *feasibility* of creating viable Computer Science minors that meet “all” of the stated minor requirements (3).

Computer Science Minor – Scenario 1 – theory oriented – electives include:

- CSCI 261 (Analysis of Algorithms)
- CSCI 262 (Introduction to Computer Science Theory) or CSCI 263 (Honors Introduction to Computer Science Theory)
- CSCI 462 (Introduction to Cryptography)
- CSCI 464 (Xtreme Theory)

Computer Science Minor – Scenario 2 – programming oriented – electives include:

- CSCI 250 (Concepts of Computer Systems)
- CSCI 251 (Concepts of Parallel and Distributed Systems)
- CSCI 344 (Programming Language Concepts)
- CSCI 541 (Programming Skills)

Computer Science Minor – Scenario 3 – intelligent systems – electives include:

- CSCI 261 (Analysis of Algorithms)
- CSCI 331 (Introduction to Intelligent Systems) (an additional prerequisite for this course is MATH 251 (Probability and Statistics I))
- CSCI 431 (Introduction to Computer Vision)
- CSCI 532 (Introduction to Intelligent Security Systems) or CSCI 539 (Seminar in Intelligent Systems)

Computer Science Minor – Scenario 4 – broad computing background (includes only one upper division course) – electives include:

- CSCI 250 (Concepts of Computer Systems)
- CSCI 251 (Concepts of Parallel and Distributed Systems)
- CSCI 261 (Analysis of Algorithms)
- CSCI 320 (Principles of Data Management) or CSCI 510 (Introduction to Computer Graphics) (an additional prerequisite for this course is MATH 241 (Linear Algebra))

The following table lists both undergraduate and graduate Computer Science courses that **could be considered for inclusion** in a Computer Science minor (4):

Undergraduate Computer Science courses	Graduate Computer Science courses
CSCI 250 Concepts of Computer Systems	CSCI 610 Foundations of Computer Graphics
CSCI 251 Concepts of Parallel and Distributed Systems	CSCI 620 Data Exploration and Management
CSCI 261 Analysis of Algorithms	CSCI 621 Database System Implementation
CSCI 262 Introduction to Computer Science Theory	CSCI 622 Secure Data Management
CSCI 263 Honors Introduction to Computer Science Theory	CSCI 630 Foundations of Intelligent Systems
CSCI 320 Principles of Data Management	CSCI 631 Foundations of Computer Vision
CSCI 331 Introduction to Intelligent Systems	CSCI 632 Mobile Robot Programming
CSCI 344 Programming Language Concepts	CSCI 633 Biologically Inspired Intelligent Systems
CSCI 351 Data Communications and Networks I	CSCI 641 Advanced Programming Skills
CSCI 352 Operating Systems	CSCI 642 Secure Coding
CSCI 420 Principles of Data Mining	CSCI 651 Foundations of Computer Networks
CSCI 431 Introduction to Computer Vision	CSCI 652 Distributed Systems
CSCI 442 Language Processors	CSCI 654 Foundations of Parallel Computing
CSCI 451 Data Communications and Networks II	CSCI 661 Foundations of Computer Science Theory
CSCI 452 System Programming	CSCI 662 Foundations of Cryptography
CSCI 453 Computer Architecture	CSCI 663 Computability
CSCI 454 Parallel Computing	CSCI 664 Computational Complexity
CSCI 455 Principles of Computer Security	CSCI 665 Foundations of Algorithms
CSCI 462 Introduction to Cryptography	CSCI 711 Global Illumination
CSCI 463 Complexity and Computability	CSCI 712 Computer Animation: Algorithms and Techniques
CSCI 464 Xtreme Theory	CSCI 713 Applied Perception in Graphics and Visualization
CSCI 510 Introduction to Computer Graphics	CSCI 714 Scientific Visualization
CSCI 519 Seminar in Computer Graphics	CSCI 715 Applications in Virtual Reality
CSCI 529 Seminar in Data Management	CSCI 719 Topics in Computer Graphics
CSCI 531 Introduction to Security	CSCI 720 Data Mining

Measurement	
CSCI 532 Introduction to Intelligent Security Systems	CSCI 721 Data Cleaning and Preparation
CSCI 539 Seminar in Intelligent Systems	CSCI 729 Topics in Data Management
CSCI 541 Programming Skills	CSCI 731 Advanced Computer Vision
CSCI 549 Seminar in Languages and Tools	CSCI 732 Image Understanding
CSCI 559 Seminar in Systems	CSCI 734 Foundations of Security Measurement and Evaluation
CSCI 569 Seminar in Theory	CSCI 735 Foundations of Intelligent Security Systems
CSCI 599 Computer Science Undergraduate Independent Study	CSCI 736 Neural Networks and Machine Learning
	CSCI 737 Pattern Recognition
	CSCI 739 Topics in Intelligent Systems
	CSCI 740 Programming Language Theory
	CSCI 742 Compiler Construction
	CSCI 746 Software Development Tools
	CSCI 749 Topics in Languages and Tools
	CSCI 753 Advanced Computer Networks
	CSCI 759 Topics in Systems
	CSCI 761 Topics in Advanced Algorithms
	CSCI 762 Advanced Cryptography
	CSCI 769 Topics in Theory

The following table lists both undergraduate and graduate Computer Science courses that are **excluded** from a Computer Science minor:

Undergraduate Computer Science courses	Graduate Computer Science courses
CMPR Computational Problem Solving for Engineers	CSCI 603 Advanced C++ and Program Design
CSCI 101 Principles of Computing	CSCI 605 Advanced Java Programming
CSCI 371 Professional Communications	CSCI 686 Graduate Professional Seminar
CSCI 499 Computer Science Undergraduate Co-op	CSCI 687 Graduate Research Seminar
CSCI 571 Honors Capstone Research	CSCI 699 Computer Science Graduate Co-op
	CSCI 788 Computer Science MS Project
	CSCI 790 Computer Science MS Thesis
	CSCI 799 Computer Science Independent Study

NOTES:

- (1) There are compelling reasons to broaden how students may be able to satisfy prerequisites. Until we teach courses under semesters, however, we will not have clear evidence as to whether this is a good idea or not. We intend to monitor all students during the first few years to see how many students start and how many students complete a CS minor. We are particularly interested in tracking any students who complete prerequisites using courses other than those identified as the standard requirement.
- (2) It's possible that in some situations (for example, with transfer students) that some students may have the required course waived. In such cases, these students will need to select an additional CS elective in place of CSCI 243.
- (3) Once the CS minor is approved, we will devise suitable information that can be posted online and given out to students who are interested in this minor. We will describe scenarios such as the ones listed and will consider adding additional scenarios over time. We will point out to students, however, that many other combinations of CS courses are viable. We expect to develop a process that students will follow initially when they sign up for a minor. As part of that process, students will need to obtain the CS minor advisor's approval for the projected courses they plan to take (as well as subsequent approval for any changes made to the original plan) and that is how we intend to ensure that students typically take at least two upper-level Computer Science electives or have approval to take just one upper-level Computer Science elective. In ***all cases***, a student who earns a CS minor will complete at least one upper-level Computer Science Elective.
- (4) There are relationships between some undergraduate and some graduate Computer Science courses and we are well aware of these. As pointed out in note (3), students will need to obtain approval for their proposed minor and we will utilize advising to make sure that students choose appropriate courses to fulfill CS minor requirements, especially when their plans include one or more graduate-level Computer Science courses.

Here is the course mask for typical prerequisites and the one CS minor required course. Students who have the necessary background could complete all prerequisites during their first year at RIT (fall and spring semesters) and would then be prepared to take CSCI 243 in the fall semester of their second year. At that point they would then need to decide which four CS Electives they planned to take in order to determine how best to schedule their remaining CS minor courses. Some students might take longer to complete the necessary prerequisites and required course, but could still have time to complete their remaining CS minor courses since many CS courses are available and offered frequently, including in the summer.

Course Number & Title	SCH	Required	Optional	Fall	Spring	Annual/Bi annual	Prerequisites
CSCI 141	4	Prerequisite		Yes	Yes	Annual	None
CSCI 142	4	Prerequisite		Yes	Yes	Annual	CSCI 141

MATH 181	4	Prerequisite		Yes	Yes	Annual	Precalculus or sufficiently high score on Math Placement Exam
MATH 182	4	Prerequisite		Yes	Yes	Annual	MATH 181 (“C” or higher)
MATH 190	3	Prerequisite		Yes	Yes	Annual	Co-requisite: MATH 172 or MATH 182
CSCI 243	3	Required		Yes	Yes	Annual	CSCI 142 or CSCI 140 or CSCI 242
Four CS Electives	12						

Total credit hours:	15 semester hours (total does not include prerequisites)
---------------------	--

As examples, let us assume that a student takes the first two years of their program to complete all the necessary prerequisite courses as well as the one required course, CSCI 243. If they were following what we identified above as scenario 1 (theory oriented), they could complete CSCI 261 and CSCI 262 in the fall and spring of their third year and then complete CSCI 464 and CSCI 462 in the fall and spring of their fourth year. If students were following scenario 2 (programming oriented), they could complete CSCI 250 and CSCI 251 in the fall and spring of their third year and then complete CSCI 344 and CSCI 541 in the fall and spring of their fourth year.

We have included lists of all potential Computer Science courses that could be used as part of a minor; details for all of these courses (including anticipated schedule of offerings and course prerequisites) are obtainable through the Semester Conversion (“Sem Con”) database, located at:

<https://sis.rit.edu/SemCon>

Minor Course Conversion Table: Quarter Calendar and Semester Calendar Comparison

Directions: The tables on this page will be used by the registrar's office to aid student's transitioning from the quarter calendar to the semester calendar.

If this minor existed in the quarter calendar and is being converted to the semester calendar please complete the following tables.

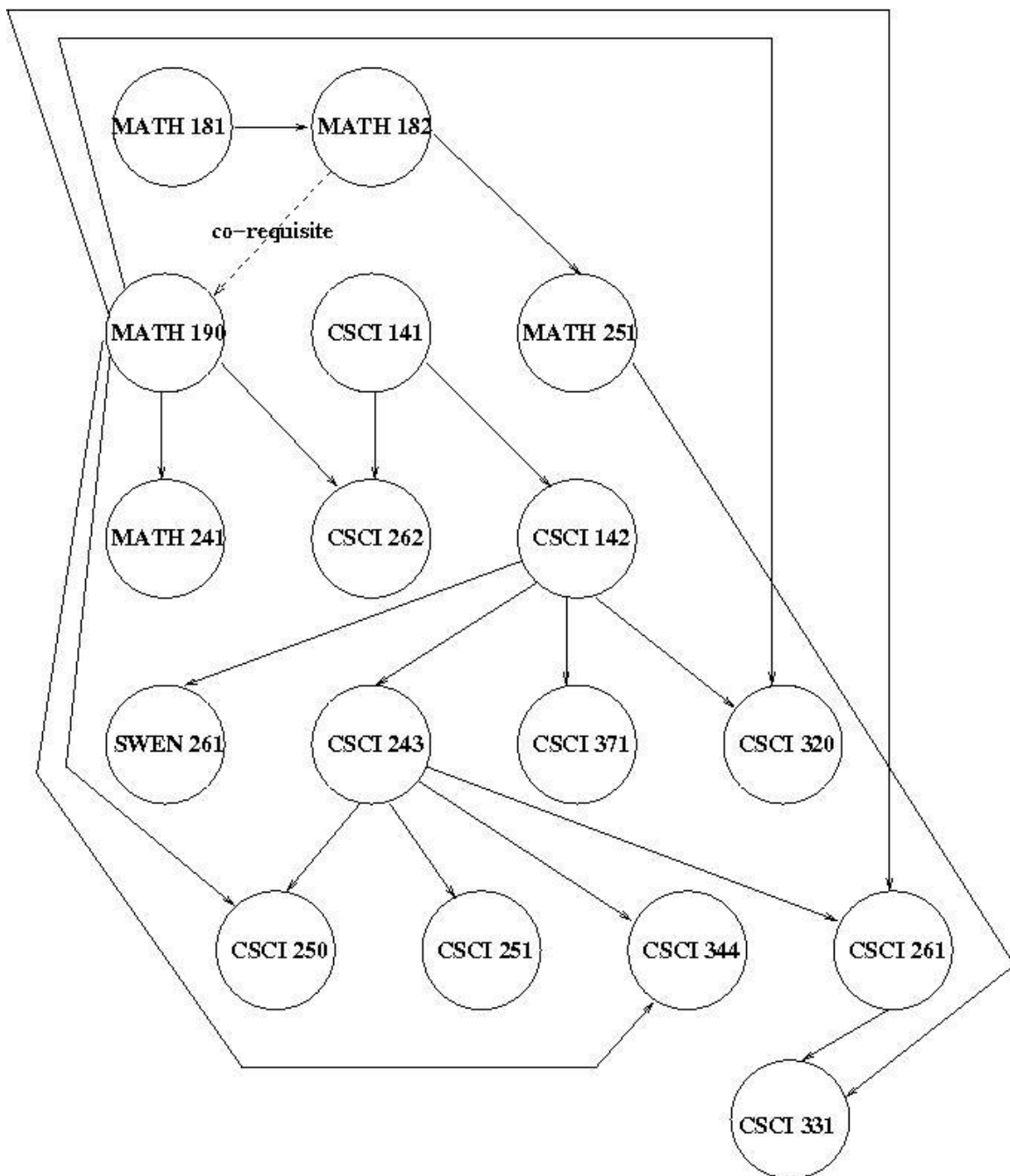
If this is a new minor that did not exist under the quarter calendar do not complete the following tables.

Use the following tables to show minor course comparison in quarter and semester calendar formats. Use courses in the (2011-12) minor mask for this table. Display all required and elective minor courses. If necessary clarify how course sequences in the quarter calendar convert to semesters by either bracketing or using some other notation.

Name of Minor in Semester Calendar:	Computer Science
Name of Minor in Quarter Calendar:	Computer Science
Name of Certifying Academic Unit:	Department of Computer Science

QUARTER: Current Minor Courses			SEMESTER: Converted Minor Courses			Comments
Course #	Course Title	QCH	Course #	Course Title	SCH	

While it is true that a minor in Computer Science does exist under quarters and there will also be a minor in Computer Science under semesters, we are not converting the existing minor under quarters to an equivalent minor under semesters. Rather, as has been pointed out throughout this document, the Computer Science minor under semesters is radically different from the Computer Science minor under quarters. Hence, we treat the minor under semesters as *“essentially a new minor”*. Thus, we do not complete the table above. We do expect to assist students who enroll in the Computer Science minor under quarters, but who complete their Computer Science minor under semesters by treating those students in the same way we treat students who complete their degrees through the transition. For each individual case, we will recommend appropriate course substitutions to allow students to complete their existing minor in a timely fashion.



Required Math/CS/SE Course Prerequisites

Required Math/CS/SE Course Number	Course Title
MATH 181	Project-Based Calculus I
MATH 182	Project-Based Calculus II
MATH 190	Discrete Math for Computing
MATH 241	Linear Algebra
MATH 251	Probability and Statistics I
CSCI 141	Computer Science I
CSCI 142	Computer Science II
CSCI 243	The Mechanics of Programming
CSCI 250	Concepts of Computer Systems
CSCI 251	Concepts of Parallel and Distributed Systems
CSCI 261	Analysis of Algorithms
CSCI 262	Introduction to Computer Science Theory
CSCI 320	Principles of Data Management
CSCI 331	Introduction to Intelligent Systems
CSCI 344	Programming Language Concepts
CSCI 371	Professional Communications
SWEN 261	Introduction to Software Engineering

Policy Name: **D1.1 MINORS POLICY**

1. Definition

A minor at RIT is a related set of academic courses consisting of no fewer than 15 semester credit hours leading to a formal designation on a student's baccalaureate transcript.

The purpose of the minor is both to broaden a student's college education and deepen it in an area outside the student's major program. A minor may be related to and complement a student's major, or it may be in a completely different academic/professional area. It is the responsibility of the academic unit proposing a minor and the unit's curriculum committee to indicate any home programs for which the minor is not a broadening experience.

In most cases, minors shall consist of a minimum of two upper division courses to provide reasonable breadth and depth within the minor.

2. Institutional parameters

- a) Minors may be discipline-based or interdisciplinary;
- b) Only matriculated students may enroll in a minor;
- c) At least nine semester credit hours of the minor must consist of courses not required by the student's home program;
- d) Students may pursue multiple minors. A minimum of nine semester credit hours must be designated towards each minor; these courses may not be counted towards other minors;
- e) The residency requirement for a minor is a minimum of nine semester credit hours consisting of RIT courses (excluding "X" graded courses);
- f) Posting of the minor on the student's academic transcript requires a minimum GPA of 2.0 in each of the minor courses;
- g) Minors may not be added to the student's academic record after the granting of the bachelor's degree.

3. Development/approval/administration processes

- a. Minors may be developed by faculty at the departmental, inter-departmental, college, or inter-college level. As part of the minor development process:
 - i. students ineligible for the proposed minor will be identified;
 - ii. prerequisites, if any, will be identified;
- b. Minor proposals must be approved by the appropriate academic unit(s) curriculum committee, and college curriculum committee(s), before being sent to the Inter-College Curriculum Committee (ICC) for final consideration and approval.
- c. The academic unit offering the minor (in the case of interdisciplinary minors, the designated college/department) is responsible for the following:
 - i. enrolling students in the minor (as space permits);
 - ii. monitoring students progress toward completion of the minor;
 - iii. authorizing the recording of the minor's completion on student's academic records;
 - iv. granting of transfer credit, credit by exam, credit by experience, course substitutions, and advanced placement;
 - v. responding to student requests for removal from the minor.
- d. As per New York State requirements, courses within the minor must be offered with sufficient frequency to allow students to complete the minor within the same time frame allowed for the completion of the baccalaureate degree.

4. Procedures for Minor revision

It is the duty of the college curriculum committee(s) involved with a minor to maintain the program's structure and coherence. Once a minor is approved by the ICC, changes to the minor that do not have a significant effect on its focus may be completed with the approval of the involved academic unit(s) and the college curriculum committee(s). Significant changes in the focus of the minor must be approved by the appropriate academic unit(s) curriculum committee(s), the college curriculum committee(s) and be resubmitted to the ICC for final consideration and approval.