NASSAU COMMUNITY COLLEGE

DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE/INFORMATION TECHNOLOGY

Course Syllabus for

CSC 130
Computer Science II

Nassau Community College Return to Campus Expectations & Guidelines

Students are required to comply with all college policies regarding vaccination, masking, social distancing, and screening or testing as outlined in the Student Guide to Returning to Campus, located on the NCC mobile app and MyNCC portal. See a summary of the college's health and safety guidelines at NCC Next (https://ncc.edu/nccnext/).

1. All students **MUST read, acknowledge, and adhere to the Student Guide to Returning to Campus via the** NCC Mobile app. Any students who will have a physical presence on campus i.e. enrolled in courses with on-campus, face-to-face ("F2F") instruction, students working on campus, student athletes, and students who plan on using on-campus resources (i.e., utilizing the campus for computer and/or internet access, or for study space access) will be responsible for complying with requirements summarized in the Guide.
2. All students should download the NCC Mobile App, which will allow them to access and complete COVID-19 Forms and Requirements for Fall 2021.
3. ***Fully vaccinated** students must electronically sign the 2021-2022 COVID-19 Attestation Form. The Attestation Form is accessible through the NCC Mobile app.
4. All students must **continue to wear masks indoors** on campus and, where practicable, observe social distancing in indoor settings.
5. **Submit Your Proof of Vaccination: Vaccine verification should be submitted** to the SUNY Vaccination Verification Application by **September 27, 2021**. **This application can also be accessed through the NCC mobile app.** More information including detailed instructions on how to upload your vaccination proof can be found on our SUNY Vaccination Mandate webpage. For information on exemptions and accommodations, please see our dedicated vaccine information site.
6. **Surveillance Testing**: Surveillance testing will be mandatory for all unvaccinated students who have been granted a medical or religious exemption and all **partially vaccinated students who have been granted a medical accommodation while they complete their vaccination requirement.
7. **Daily Screening:** Daily screening and symptom monitoring will continue for all unvaccinated students who have been granted a medical or religious exemption and all partially vaccinated students who have been granted a medical accommodation while they complete their vaccination requirement. Screening must be completed each day before coming to campus.

8. QR Codes and Contact Tracing: As you move around campus, you will be encouraged to check-in to buildings and rooms by scanning QR codes posted on doors and walls prior to entering. In order to do so you must download the NCC mobile app. Check-in by clicking the box with the line through it at the top left of the NCC mobile app and scanning the QR code in the room. **DO NOT SCAN THE CODE WITH YOUR PHONE'S CAMERA APP. THEY MUST BE SCANNED IN THE NASSAU APP TO CHECK-IN PROPERLY.** Be sure to look for and scan QR codes in every area you visit on campus. In the case of a positive COVID case on-campus, this will assist the College in our contact tracing efforts. Be sure to scan the codes again as you leave the room to check-out.

Course Information

- Title Computer Science II
- Credit Hours 4 Credits
- Number CSC 130
- Section EA1
- CRN 13202
- Semester Fall 2021
- Meeting time Tuesday, Thursday 5:30pm - 8:20pm
- Location Remote via Zoom

Instructor/Contact Information

- Name Franklin Graham
- Office location B3041
- Office hours Monday - Thursday 4:00pm - 4:30pm
- Office telephone 572-7383 ext. 26807
- Email address Franklin.Graham@ncc.edu
- Blackboard link https://ncc.sln.suny.edu
- Website http://matcmp.ncc.edu/grahamf/
- Student email http://www.ncc.edu/studentemail

Course Description

- CSC 130: Computer Science II

- Prerequisites*:* At least a C in CSC 120 (previously CMP 210).

- Description*:* This course expands on the design of object-oriented programs introduced in Computer Science I. Students will be introduced to abstract data types including stacks, queues, and lists. Emphasis is placed on the design and implementation of these abstract data types as well as applications that utilize them. Object-oriented programming concepts such as composition, inheritance, polymorphism, and exception handling are utilized throughout this course. An introduction to binary-trees and recursion is provided. Includes supervised hands-on laboratory component. (4 lecture, 6 contact hours) Laboratory fee applies.

- CSC 130 satisfies NCC GEN EDMATH. CSC 130 serves as a prerequisite for CSC 217 (C Programming Language), CSC 230 (Computer Science 2), CSC 240 (Mobile Application Development), and CSC 260 (Computer Organization and Architecture).

- Detailed Topics Outline (chapter information will be updated after the eText content gets sorted)

Topic / Subtopic	Text Chapter(s)	# Class Meetings*	# Laboratory Meetings **
1. Object oriented programming review	1, 11 – 18 (CSC 120 Content)	1	1
2. Inheritance, polymorphism abstract classes, interfaces	3, 5 Lecture Notes	2	1
3. Exception handling, File I/O	2, 4 Lecture Notes	2	1
4. Array-based implementation of basic abstract data types (stack, queue, list)	8.1, 8.6, 8.9, 8.10 Lecture Notes	8	4
5. Linked lists	8	1	
6. Linked implementation of basic abstract data types (stack, queue, list)	8 Lecture Notes	6	3
7. Java Collections Framework	7 Lecture Notes	2	
8. Recursion	9 Lecture Notes	2	1
9. Binary Trees and Recursive implementation of the binary search tree ADT	10 Lecture Notes	2	1
10. Review & Testing		4	3
Total Hours		**30**	**15**

* Lecture time represents a regular 1 ¼ hour class period.
** Laboratory Meetings are 2.75 hours in length with a 15-minute break.
Students are encouraged to take a break whenever it is most appropriate for them.

Learning Outcomes and Objectives

General objectives: To expand on the student's knowledge of object-oriented, high level programming concepts and techniques and to refine their problem solving skills.

Specific Learning Goals & Outcomes

Learning Goal 1 - Stack Operations Students should understand how to utilize basic stack operations to solve a given problem.	
Learning Outcome	**Mapping**
Learning Outcome 1.1 - Programming a Stack	**Computer Science, A.S. Outcome Set:** 2.3 Write New Code
Students should be able to write a method that pushes and pops a stack to achieve a specified goal.	
Learning Goal 2 - Linked List Organization Students should understand how a linked list is organized.	
Learning Outcome	**Mapping**
Learning Outcome 2.1 - Programming a Linked List	**Computer Science, A.S. Outcome Set:** 2.2 Modify Existing Code
Students should be able to write methods that modify linked lists.	
Learning Goal 3 - Queue Organization Students should understand how a circular array based queue is organized.	
Learning Outcome	**Mapping**
Learning Outcome 3.1 - Understanding a Queue	**Computer Science, A.S. Outcome Set:** 2.1 Interpret Existing Code
Students should be able to trace a code segment of a *CircularArrayQueue* by specifying what is stored in memory and the output produced.	
Learning Outcome	**Mapping**
Learning Outcome 3.2 - Choosing a Stack or Queue	**Computer Science, A.S. Outcome Set:** 1.2 Data Structures
Students should determine which data structure (a stack or a queue) is most appropriate for a given situation.	
Learning Goal 4 - Recursive Programming Students should understand how recursion works in a program.	
Learning Outcome	**Mapping**
Learning Outcome 4.1 - Tracing Recursive Functions	**Computer Science, A.S. Outcome Set:** 2.1 Interpret Existing Code
Students should be able to trace recursive functions.	
Learning Outcome 4.2 - Understanding Recursion	**Computer Science, A.S. Outcome Set:** 1.1 Terminology
Students should demonstrate a broader understanding of recursion	
Learning Goal 5 - Tree Data Structures Students should understand tree data structures	
Learning Outcome	**Mapping**
Learning Outcome 5.1 - Draw a Binary Search Tree	**Computer Science, A.S. Outcome Set:** 3.2 Recursive Algorithms
Students should determine the shape of a binary search tree and indicate the output produced by a postorder traversal.	

Instructional Methods

This course is taught using a variety of instructional methods including lecture, small group work when applicable, project creation, and an official 2.5-hour laboratory component. During lab time it is expected that students work on directed laboratory assignments to aid in their understanding of the course material. Instructors must be present for these laboratory sessions and should provide the students with written instructions for each assignment.

Textbook and Materials

zyBook: CSC 130: Computer Science II
zyBook code: NCCCSC130GrahamFall2021
zyBook ISBN: 979-8-203-90456-0

- **Compiler** - Students should use Eclipse as the integrated development environment (IDE)/editor for all programming assignments. Eclipse 2021-06 R includes a JRE for Windows, macOS, and Linux.

Student Responsibilities /Course Policies

- **Projects (*All class files must be well documented as specified in Lab 0*)**
  There will be **5** programming projects. All projects are expected to be handed in on time. Students are required to submit programs that are syntax free, and produce some output, even if the output is incorrect. Programs which contain syntax errors will receive a 0. Students must submit **a Zip file**, via Blackboard, containing the **entire Eclipse project** by **11:59pm** of the day the assignment is due.
- **Labs (*All class files must be well documented as specified in Lab 0*)**
  There will be **12** laboratory assignments. Students will work in pairs on the assignment distributed at each of these meetings, and each student will be responsible for completing and submitting these assignments **by 11:59 pm on the Saturday** of that week. Due to the fact that you will be working in pairs, it is essential that you be present and on time for all laboratory meetings.
- **Exams**
  There will be **2 exams** given throughout the semester in addition to a **final exam** all on Blackboard. You must be visible on Zoom during the exam. Make-up exams are normally not given, but consideration will be given to those students who contact me before the exam (via e-mail or Zoom) and provide a valid, documented reason for missing the exam.
- **Attendance**
  Attendance is a critical aspect of this course. As such, attendance will be taken at every class meeting. Students are expected to be in class on time and stay for the duration of the scheduled time. Students are responsible for all material missed due to absence and should contact me or another student prior to the next scheduled class meeting to determine what was covered and/or assigned. Any student absent on the day an assignment is due is still responsible for submitting the assignment electronically by the specified deadline. Videos will not be made available.
- **Withdrawal Policy**
  I will grant a grade of "W" any time before the final. It is your responsibility to make a formal request, if you wish to withdraw from this class. This option is available in the NCC Portal.
  If you stop attending and do not file an official withdrawal form, a grade of "UW" (Unofficial Withdrawal) will be assigned. A grade of "UW" will count toward your GPA as an F.

- **Academic Dishonesty & Plagiarism**

  Academic dishonesty, which includes plagiarism and cheating, will result in some form of disciplinary action that may lead to suspension or expulsion under the rules of the Student Code of Conduct. Cheating can take many forms including but not limited to copying from another student on an examination, using improper forms of assistance, or receiving unauthorized aid when preparing an independent item of work to be submitted for a grade, be it in written, verbal or electronic form. Anyone who assists or conspires to assist another in an act of plagiarism or any other form of academic dishonesty may also be subject to disciplinary action.

  Plagiarism is a particular type of academic dishonesty that involves taking the words, phrases or ideas of another person and presenting them as one's own. This can include using whole papers and paragraphs or even sentences or phrases. Plagiarized work may also involve statistics, lab assignments, art work, graphics, photographs, computer programs and other materials. The sources of plagiarized materials include but are not limited to books, magazines, encyclopedias or journals; electronic retrieval sources such as materials on the Internet; other individuals; or paper writing services.

  A student may be judged guilty of plagiarism if the student:

  (a) Submits as one's own an assignment produced by another, in whole or in part.

  (b) Submits the exact words of another, paraphrases the words of another or presents statistics, lab assignments, art work, graphics, photographs, computer programs and other materials without attributing the work to the source, suggesting that this work is the student's own.

  Allegations of student plagiarism and academic dishonesty will be dealt with by the appropriate academic department personnel. It is the policy of Nassau Community College that, at the discretion of the faculty member, serious acts will be reported in writing to the Office of the Dean of Students, where such records will be kept for a period of five years beyond the student's last semester of attendance at the College. These records will remain internal to the College and will not be used in any evaluation made for an outside individual or agency unless there is a disciplinary action determined by a formal ruling under the Student Code of Conduct, in which case only those records pertaining to the disciplinary action may apply. A student whose alleged action is reported to the Office of the Dean of Students will be notified by that office and will have the right to submit a letter of denial or explanation. The Dean will use his/her discretion in determining whether the alleged violation(s) could warrant disciplinary action under the Student Code of Conduct. In that case the procedures governing the Code of Conduct will be initiated.

- **Copyright statement**

  The Higher Education Opportunity Act of 2008 (HEOA) requires the College to address unauthorized distribution of copyrighted materials, including unauthorized peer-to-peer file sharing.

  Thus, the College strictly prohibits the users of its networks from engaging in unauthorized distribution of copyrighted materials, including unauthorized peer-to-peer file sharing. Anyone who engages in such illegal file sharing is violating the United States Copyright law, and may be subject to criminal and civil penalties. Under federal law, a person found to have infringed upon a copyrighted work may be liable for actual damages and lost profits attributable to the infringement, and statutory damages of up to $150,000. The copyright owner also has the right to permanently enjoin an infringer from further infringing activities, and the infringing copies and equipment used in the infringement can be impounded and destroyed. If a copyright owner elected to bring a civil lawsuit against the copyright infringer and ultimately prevailed in the claim, the infringer may also become liable to the copyright owner for their attorney's fees and court costs. Finally, criminal penalties may be assessed against the infringer and could include jail time, depending upon the

severity of the violation. Students should be aware that unauthorized or illegal use of College computers (such as engaging in illegal file sharing and distribution of copyrighted materials), is an infraction of the Student Code of Conduct and may subject them to disciplinary measures. To explore legal alternatives to unauthorized downloading, please consult the following website: http://www.educause.edu/legalcontent.

Course Resources

- **Web sites**
  Students will use email and Blackboard (an online course management tool) throughout this course. Students are expected to read their NCC email accounts (http://www.ncc.edu/studentemail) daily and to check **Blackboard** (https://ncc.sln.suny.edu) for notification of assignments and to submit their work for grading.

- **Labs and learning centers: COMPUTER CENTER REQUIREMENT**
  As part of this course, students should avail themselves of further study and/or educational assistance available from faculty in the Computer Learning Center. Utilizing this resource is deemed an integral part of the course and will help the student master necessary knowledge and skills.

Assessments and Grading Methods

- Final grades will be determined by the following percentages:
  - Exams 1 and 2 ..... 40% (20% each)
  - Final Exam 22%
  - 5 Projects ..... 26% (projects 1, 2, 3, 5 - 5% each, project 4 - 6%)
  - 12 Laboratory Assignments ..... 12% (1% each)

Americans with Disabilities Statement & Non-Discrimination Statement

- If you have a physical, psychological, medical, or learning disability that may have an impact on your ability to carry out the assigned coursework, I urge you to contact the staff at the **Center for Students with Disabilities (CSD)**, Building U, (516)572-7241, TTY (516)572-7617. The counselors at CSD will review your concerns and determine to what reasonable accommodations you are entitled as covered by the Americans with Disabilities Act and section 504 of the Rehabilitation Act of 1973. All information and documentation pertaining to personal disabilities will be kept confidential.

Course Schedule and Important Dates

- The following is intended to provide you with an outline of how this course will progress. Dates of exams, project assignments, and project due dates may be adjusted to account for progress of the class as a whole. **The *Laboratory* component of this course is critical and is designed to re-enforce material presented in the classroom.**

Week	Date	Topic
1	9/02	Introduction, course outline and policies; review of CS1 topics data types, classes, objects, methods, arrays, and encapsulation
2	**9/07**	***No Class - Rosh Hashanna***
	9/09	Arrays, *Lab #0 – arrays, documenting code*
3	9/14	Inheritance, Abstract Classes, Interfaces, Polymorpism
	9/16	***No Class - Yom Kippur***
4	9/21	*Lab# 1 - 2-Dimensional arrays (Matrix class)*
	9/23	Exception handling, Introduction to **Stacks** using an array-based implementation, introduction to generics
5	9/28	*Lab #2 – Inheritance, Abstract Classes, Interfaces, Polymorpism*
	9/30	Introduction to linked structures, Linked implementation of stacks
6	10/05	*Lab #3 – File I/O & Exception handling*, Review for exam 1
	10/07	**Exam 1**
7	10/12	*Lab #4 – array-based Stack and LinkedStack classes*
	10/14	**Queues**: an array-based implementation
8	10/19	*Lab #5 – The CircularArrayQueue class*
	10/21	**Linked Queues**
9	10/26	*Lab #6 – The LinkedQueue class*
	10/28	**Unordered Lists**: an array-based implementation
10	11/02	**Unordered Lists** cont., exam 2 review
	11/04	**Exam 2**
11	11/09	*Lab #7 – Array-based unordered lists*
	11/10	**Ordered Lists**: Manipulating an ordered list, searching, sorting an array-based implementation
12	**11/16**	***No Class - Evening Activity***
	11/18	*Lab #8 – Array-based ordered lists*
13	11/23	Unordered Lists: a linked implementation
	11/25	***No Class - Thanksgiving***
14	11/30	*Lab #9 – Unordered lists using linked lists*
	12/02	Introduction to **Recursion**
16	12/07	*Lab #10 – Recursion*
	12/09	Introduction to **Binary Trees** and **Binary Search Trees**
17	12/14	*Lab #11 – Binary search trees*
	12/16	Binary search trees: additional operations, Review for final exam,
18	**12/21**	**Final Exam**

Please note all dates and order of content may be modified as the course progresses.

ACADEMIC CALENDAR
FALL 2021

SEPTEMBER

1	**Wednesday**	**Fall, 1st half, Online Education & Evening classes begin**
		Late payment fee begins
3	**Friday**	**Evening classes do not meet** (classes beginning AFTER 5:01 p.m.)
4	**Saturday**	**Classes do not meet**
5	**Sunday**	**Classes do not meet**
6	**Monday**	**Labor Day – classes do not meet – COLLEGE HOLIDAY**
7	**Tuesday**	**Rosh Hashanah – classes do not meet**
		75% refund ends online by 11:59 p.m.
		1st half classes last day drop without a W grade online by 11:59 p.m.
8	Wednesday	Fall & 1st half classes last day drop/add
14	Tuesday	50% refund ends online by 11:59 p.m.
15	**Wednesday**	**Evening classes do not meet** (classes beginning AFTER 5:01 p.m.)
16	**Thursday**	**Yom Kippur – classes do not meet – COLLEGE HOLIDAY – offices closed**
21	Tuesday	25% refund ends online by 11:59 p.m.
		Fall classes last day drop without a W grade online by 11:59 p.m.

OCTOBER

1	**Friday**	**Immunization Records Submission Deadline**
4	Monday	Tuition Payment Plan – second payment due
12	Tuesday	Evening Activity Hour: 8:30 p.m. class will not meet;
		all other classes follow a regular schedule
25	Monday	Deadline for Fall graduation application

NOVEMBER

4	Thursday	Tuition Payment Plan – third and final payment due
5	**Friday**	**Fall last day automatic W**
10	Wednesday	**DAY & EVENING classes meet on a THURSDAY schedule**
11	**Thursday**	**Veterans' Day – classes do not meet; COLLEGE HOLIDAY**
16	Tuesday	Evening Activity Hour: 7:00 p.m. class will not meet;
		all other classes follow a regular schedule
24	Wednesday	Evening classes do not meet (classes beginning AFTER 5:01 p.m.)
25	**Thursday**	**Thanksgiving – classes do not meet; COLLEGE HOLIDAY**
26	**Friday**	**Thanksgiving Recess – classes do not meet; COLLEGE HOLIDAY**
27	**Saturday**	**Classes do not meet**

DECEMBER

7	Tuesday	Evening Activity Hour: 5:30 p.m. classes will not meet;
		all other classes follow a regular schedule
19	Sunday	Weekend classes end
20	Monday	EVENING classes meet on a WEDNESDAY schedule
21	**Tuesday**	**All 3 credit Evening classes must be extended by 5 minutes for final exams**
		Evening classes end
22	Wednesday	DAY classes meet on a THURSDAY schedule
		Fall, 2nd half & Online Education classes end
		ME Makeup – if necessary EVENING classes will meet
23	Thursday	MD Makeup – if necessary DAY classes will meet

See before links to the complete academic calendars:
https://www.ncc.edu/aboutncc/cal_pdfs/fall.pdf
https://collegecatalog.ncc.edu/_resources/pdfs/AcademicCalendar.pdf

Programming Grading Rubric

Submission Criteria

- Students must submit the exported Eclipse project containing the *.src folder*, and the *.classpath* and *.project* files by the due date. The zip files must be in the form *csc130.LastNameProject#* for programming projects and *csc130LastnameLab#* for lab assignments. Students should read the assignments, as other directions will be included.

Programming Style Criteria

- The submitted code must be readable, neat, and well organized. Students should use indentation consistently in compliance with standard practices. White space (spaces tabs, blank lines) should be used to make the code readable. If in doubt, students should use the code format option in Eclipse.

- Except for for…loop variables, variables and methods should be given meaningful names to help the reader understand their purpose, and they should comply with standard programming practice – camelCase.

- Commented, and therefore not executable, code should be removed prior to submission.

Program Documentation Criteria

- **Every file in the project should start with a header comment (Java docs style), as specified in Lab 0**. The comment should contain at least a description of what the code does, the author, or authors in the case of lab assignments, the name of the file, the date the assignment is due and the date it was completed. In the case of project assignments, if the student received help from other sources – the Computer Learning Center – that information should be included.

- Comments should be placed above methods as necessary for use with the Javadoc tool (the purpose of the method, @param, @return, and @throws tags should be used - see http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html.)

- Statements should be commented. Obviously, every line does not need to be commented. However, a complex statement should be commented to help the reader understand its purpose.

Program Correctness Criteria

- This is the most important criterion and therefore carries the most weight. The project should meet the specifications and produce the expected results/output. Even though the project produces the desired results, appropriate data structures should be used, and the code used to generate these results should be efficient. Programs that take too much time to execute, because of a poor algorithm choice, or have too many unnecessary lines, when a simpler solution with less code could be used, will be deemed inefficient.

  Exceptions should be handled to produce graceful exists; programs should not end abruptly (crash) because an unhandled or poorly handled exception occurred.

- In addition, the output should be formatted. Use the *String.format* method or the *System.out.format* method to format numbers (https://docs.oracle.com/javase/tutorial/java/data/numberformat.html).

Programming Grading Rubric

Submission Criteria	
Correct project submission zip file, correctly-named project (as specified) in zip file, on time	5%
Programming Style Criteria	
Appropriate choice of variable names and data types	5%
Readability/Neatest (code indentation)	5%
Program Documentation Criteria	
Header comment (Description, author, date due, etc.)	5%
Documentation (class and method documentation as necessary)	10%
Program Correctness Criteria	
Correct algorithm/logic, computes without run-time errors	15%
Correct results	40%
Output formatting quality	5%
Efficiency	10%

Note: no points will be awarded if the program has syntax errors or is completely incorrect and does not meet the specifications, even if it is well documented and well organized.

2% will be deducted for each day assignment is late.

Criterion	Unsatisfactory (0%) 0	Poor (20 - 40%) 1 - 2	Adequate (60%) 3	Good (80%) 4	Excellent (100%) 5
Submission 5%	Incorrect or no zip file submitted, incorrect project name, missing .project and .classpath	Incorrect zip file, or project name is incorrect. [.project and .classpath files (2)]	Correct zip file and project name, but missing .project and .classpath files	Not possible	Correct project zip file, .project and .classpath files, and project name is correct.
Programming Style 10%	Inappropriate choice of variable names and data types; code is difficult to read and poorly organized.	At least one major problem with choice of variable names and data types, use of whitespace, indentation, or layout.	Minor inconsistencies in use of variable names, data types, indentation, and layout.	Appropriate choice of variable names and data types; minor inconsistencies in indentation.	Appropriate choice of variable names and data types, and code is easy to read and well-organized.
Program Documentation Header - 5% Class, method, statement – 10%	No header, class, method, or statement comments.	Class header or method comments missing. Complicated code not commented.	Class header and method comments included, but comments missing or somewhat useful in understanding the code.	Class header and method comments included but a few comments are missing.	Code is well documented and commented as specified.
Correct results 40%	Program produces no correct results at all. **Demonstrates limited or no understanding of concepts.**	Programs produces a few correct results but does not check for errors. **Demonstrates some understanding of concepts.**	Programs produces some correct results, may fail for minor special cases; does little error checking. **Demonstrates an adequate understanding of concepts.**	Program works for typical input, and does some error checking. **Demonstrates a good understanding of concepts.**	Program uses an appropriate algorithm and produces correct results in all cases and does error checking. **Demonstrates thorough understanding of concepts.**

Correct algorithm/logic 15%	Wrong algorithm and incorrect data structures used.	Program uses a somewhat correct algorithm and data structures, but always produces runtime errors.	Program uses the correct algorithm and some correct data structures, but sometimes produces runtime errors.	Program uses the correct algorithm and data structures but produces a minor runtime error.	Program uses the correct algorithm and data structures, and never produces runtime errors.
Efficiency 10%	Very inefficient with no effort to eliminate unnecessarily long and repetitious code.	The code is somewhat efficient with some attempt to eliminate repetitious code.	The code is somewhat efficient with little repetitious code.	The code is mostly efficient with little repetitious code.	Very efficient and has no repetitious code.
Output formatting 5%	Output poorly organized and not formatted at all.	Output well organized but has major inconsistencies in formatting.	Output well organized but has a few minor formatting issues.	Output well organized but has 1 or 2 minor formatting issues.	Output formatted well and very well organized.