Creating Grading and Feedback System of a Console Based Time Scheduling of College Students System Utilizing Data Structures and Algorithm

**Subtitle focusing on your topic**

Randy, Sarte, Camposo[[1]](#footnote-1)\* (Student)

First author's affiliation, an Institution with a very long name, xxxx@gmail.com

MAX ANGELO DAPITILLA PERIN (INSTRUCTOR)

Faculty of Department of Computer Science, College of Technology and Allied Sciences, Bohol Island State University-Bilar Campus, maxangelo.perin@bisu.edu.ph

Although there is no distinctive header, this is the abstract. This submission template allows authors to submit their papers for review to an ACM Conference or Journal without any output design specifications incorporated at this point in the process. The ACM manuscript template is a single column document that allows authors to type their content into the pre-existing set of paragraph formatting styles applied to the sample placeholder text here. Throughout the document you will find further instructions on how to format your text. If your conference’s review process will be double-blind: The submitted document should not include author information and should not include acknowledgments, citations or discussion of related work that would make the authorship apparent. Submissions containing author identifying information may be subject to rejection without review. Upon acceptance, the author and affiliation information must be added to your paper.

CCS CONCEPTS • Theory of computation • Design and analysis of algorithms• Data structures design and analysis

Additional Keywords and Phrases: Time Scheduling and Management, Data Structures, Algorithmic Optimization, Grading and Feedback.

ACM Reference Format:

**Randy Sarte Camposo**, MAX ANGELO DAPITILLA PERIN. 2023. **Creating Grading and feedback System of a Console Based Time Scheduling of College Students System Utilizing Data Structures and Algorithm**. In Research Project Presentation for Bachelor of Science in Computer Science 3 in CS 212 – Data Structures and Algorithm S.Y. 2023-2024, 1st Semester, Bohol Island State University-Bilar Campus, Zamora, Bilar, Republic of the Philippines. ACM, New York, NY, USA

1. INTRODUCTION

Creating a grading and feedback System for a console-based time scheduling system for college students is a vital and challenging task, and it plays a pivotal role in enhancing the overall educational experience. In today's fast-paced world, students need effective tools to manage their time efficiently and make the most of their college years.[1]This project aims to address this need by utilizing data structures and algorithms to design a robust and user-friendly scheduling system, accompanied by a grading and feedback component.

[2]The proposed system will employ data structures such as linked lists, arrays, and trees to manage student schedules efficiently. Algorithms will be used to optimize scheduling, taking into account factors like course prerequisites, student preferences, and available resources. The grading and feedback system will provide a means for instructors to assess student performance and for students to receive constructive feedback on their academic progress. [3]This system will not only streamline scheduling but also promote a culture of continuous improvement and accountability among college students. In this project, we will explore the design, implementation, and integration of these components, ensuring that they work seamlessly to enhance the educational experience and empower students to reach their full potential.

* 1. Console-based Creating a Grading Feedback System

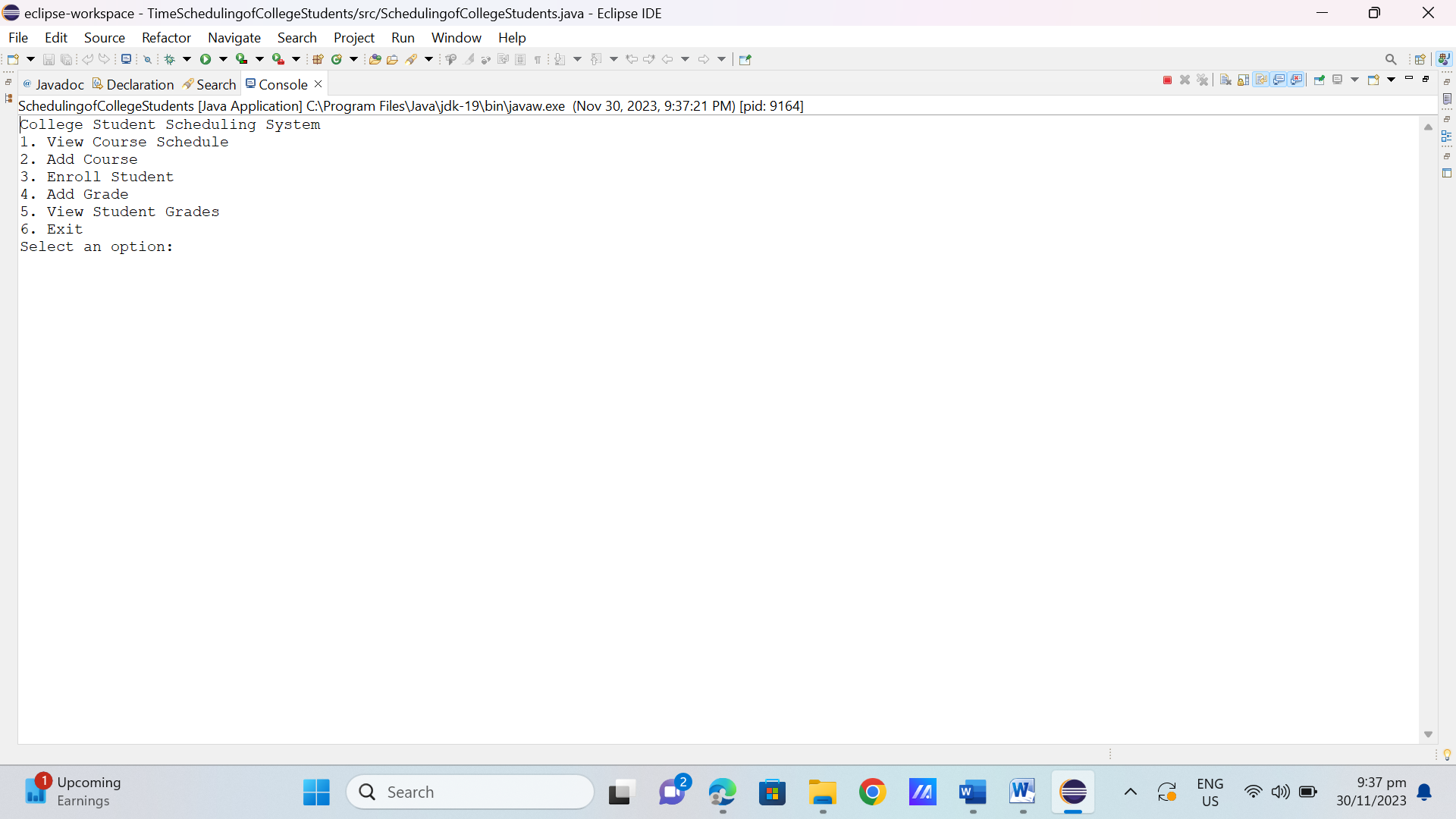


Figure1. Creating a different choices of grading feedback System

* 1. More about the submission template

Thissubmission version of your paper should not have headers or footers, these will be added when your manuscript is processed after acceptance. It should remain in a one-column format—please do not alter any of the styles or margins.

*If a paper is accepted for publication*, authors will be instructed on the next steps. Authors must then follow the submission instructions found on their respective publication’s web page. Once your submission is received, your paper will be processed to produce the formatted Word, PDF, and HTML5 output formats, which will be provided to you for review, revision/resubmission (if applicable), and approval.

* 1. Inserting CCS concepts

The new template enables you to import required indexing concepts for your article from the [ACM Computing Classification System (CCS)](http://www.acm.org/publications/class-2012) using an [indexing support tool](http://dl.acm.org/ccs/ccs.cfm?) found in the ACM Digital Library (DL). The tool generates formatted text after you have selected your terms. To insert CCS terms into your document, copy and paste the formatted text from the CCS tool using the “<https://dl.acm.org/ccs/ccs.cfm>” link into the “CCS CONCEPTS” section.

An additional step is necessary to ensure that the proper CCS terms are added to the Digital Library citation page: from the “view CCS TeX Code” listing, click on “Show the XML Only.” Highlight and copy the XML code from the window. You must insert the XML code into your Word document’s properties: from your Word document, click on “**File**”, then click on the “**Info**” tab on the left-hand side panel, then click “**Properties**” and select “**Show All Properties.**” Click within the “Comments” metadata field and paste the XML data.

1. PROPOSED METHODOLOGY

Provide an overview of the importance of a grading and feedback system in educational settings. Develop a comprehensive methodology for implementing an effective grading and feedback system. And the first thing that I do is to Identify stakeholder needs and system requirements for grading and feedback processes and we can see the different choices, and each choices haves a capability to give a feedback of your choices. Hence it helps the students to View their Schedule and see if they are being enrolled of that subject.

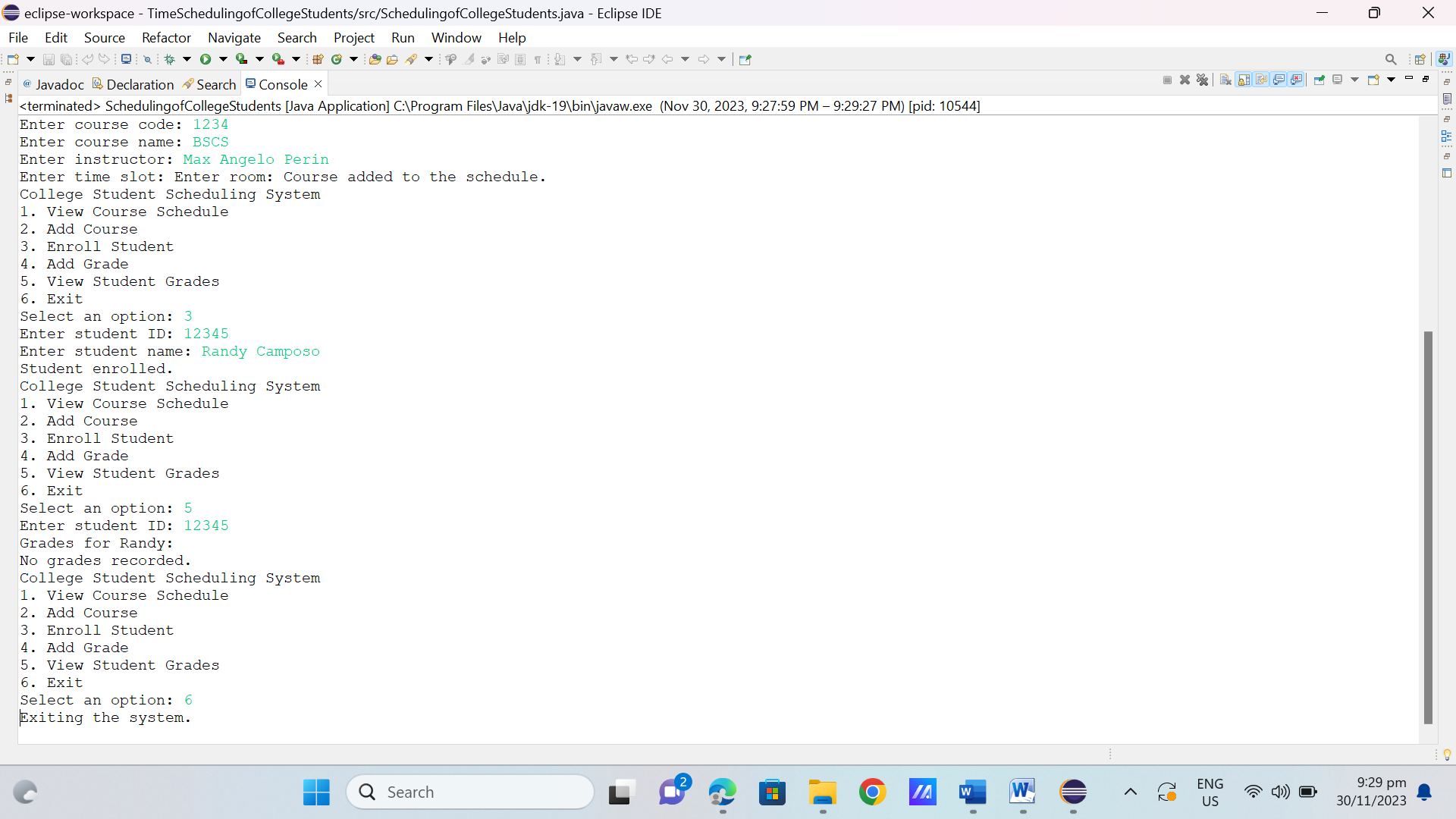


Figure2. Showing all of the choices of the Grading Feedback System.

* 1. Tables

Tables are “float elements” which should be inserted after their first text reference and have specific styles for identification. Do not use images to present tables, or they will be inaccessible to readers using assistive technologies.

Authors can insert tables by using the M.S. Word option (INSERT ->Table) and providing the required row and column size. Every table must have a caption (title) above it, which must have the **“TableCaption**” style applied. Please note that tables **should not** be supplied as image files, but if they are images they must have the “Image” style applied. As an example, Table 1 shows all the styles available in this template, to be applied to the respective element of your text.

Table 1: Styles available in the Word template

| Style Tag | Definition | Style Tag | Definition |
| --- | --- | --- | --- |
| Title\_document | main title of article | ListParagraph | list items |
| Subtitle | subtitle of article | Statements | math statements |
| Authors | author name | Extract | block quotations |
| Affiliation | author affiliation information | Algorithm Caption | caption for algorithm |
| AuthNotes | footnote to author(s) | AckHead | heading for acknowledgements |
| Abstract | abstract text | AckPara | acknowledgements text |
| CCSHead | heading for CSS Concepts | GrantSponsor | sponsor of grant |
| CCSDescription | CSS terms | GrantNumber | number for the grant |
| KeyWordHead | heading for keywords | ReferenceHead | heading for references |
| Keywords | keywords text | Bib\_entry | references |
| ORCID | author's ORCHID # | AppendixH1 | appendix heading level 1 |
| Head1 | heading level 1 | AppendixH2 | appendix heading level 2 |
| Head2 | heading level 2 | AppendixH3 | appendix heading level 3 |
| Head3 | heading level 3 | TableCaption | title of table |
| PostHeadPara | first paragraph after a heading | TableHead  TableFootnote | column head of table  footnote to table |
| Para | Subsequent paragraphs of general text | Image | figures |
| ParaContinue  DisplayFormula | flush left text after display items like math equations, lists etc.  numbered math equation | DOI | Digital object identifier |
| DisplayFormulaUnnum | unnumbered equations | Label | labela |
| ComputerCode | Display Computer codes | In-text code | intext computer code |
| Short Title | Short title of article |  |  |

a This is example of table footnote.

Tables can be very difficult for people using screen reader technology to understand unless they include markup that explicitly defines the relationships between all the parts (i.e.: headers and data cells). *A key to making data tables accessible to screen reader users is to clearly identify column and row headers.* In Word, authors should identify which row or rows contain column headers. Below are the steps to do this:

1. Select that table’s row, then right-click the row and select “Table Properties”;
2. In the *Table Properties* window, click the *Row* tab and select the box that says “Repeat as header row at the top of each page.”

Or

Apply the “table head” style by highlighting the respective row and applying the “**TableHead**” style found in the “Body Element” section of the ACM Master Article Template.

* 1. Figures

Figures are “float elements” which should be inserted after their first text reference, and have specific styles for identification. Insert a figure and apply the “**Image**” paragraph style to it. For the figure caption, apply the style “**FigureCaption.**”

To accommodate readers with color vision differences, figures should still be usable when printed in grayscale. Refer to elements of the figure with non-color terms, for example “indicated as squares” instead of “indicated in blue”. Use different patterns in bar charts, different line patterns in graphs, and different shapes in plots to distinguish groups of elements and reinforce color differences.

* + 1. Half Width Figures.

Figure 1 is an example of a figure and caption spanning the half-page width (one column in a two column format) with the styles applied. If your figure contains third-party material, you must clearly identify it as such, as shown in the example below.



Figure 1: 1907 Franklin Model D roadster. Photograph by Harris & Ewing, Inc. [Public domain], via Wikimedia Commons. (https://goo.gl/VLCRBB)

* + 1. Full Width Figures.

Figure 2 is an example of a figure and caption spanning the full-page width with the styles applied. If your figure contains third-party material, you must clearly identify it as such, as shown in the examples.



Figure 2: Mockup of a bombe machine at Bletchley Part. Photograph by Sarah Hartwell. [Public domain], via Wikimedia Commons. (<https://commons.wikimedia.org/wiki/File:TuringBombeBletchleyPark.jpg>)

* + 1. Multi-part figure.

Authors can also insert a multi-part figure above a single caption. Every inserted figure must have the “Image” style applied. Below are instructions regarding how to insert a multi-part figure in your paper.

* If the author wants to insert two multi-part images, they must draw a one row and one column table and insert the images one-by-one in the cells.
* If the author wants to insert three multi-part images, they must draw a one-row and three-column table and insert the images one by one in all three cells.
* If the author wants to insert four multi-part images, they must draw a two-row and two-column table and insert the images one-by-one in all four cells. (see the following example):

| Figure 2: The layout of multipart images should be as per the above example within the table in image 1. | Figure 2: The layout of multipart images should be as per the above example within the table in image 2. |
| --- | --- |
| Figure 2: The layout of multipart images should be as per the above example within the table in image 3. | Figure 2: The layout of multipart images should be as per the above example within the table in image 4. |

Figure 3: The layout of multipart images should be as per the above example within the table. All images must have the “Image” style applied.

* + 1. Figure Descriptions.

Every figure should have a figure description unless it is purely decorative. These descriptions convey what’s in the image to someone who cannot see it. They are also used by search engine crawlers for indexing images, and when images cannot be loaded.

A figure description must be unformatted plain text less than xxx characters long. Figure descriptions should not repeat the figure caption – their purpose is to capture important information that is not already provided in the caption or the main text of the paper. For figures that convey important and complex new information, a short plain text description may not be adequate. More complex alternative descriptions can be placed in an appendix and referenced in a short figure description. For example, provide a data table capturing the information in a bar chart, or a structured list representing a graph. For additional information regarding how best to write figure descriptions and why doing this is so important, please see [https://www.acm.org/accessibility.](https://www.acm.org/accessibility)

The instructions below describe the required steps authors need to follow in order to insert descriptive text for figures (alt-txt value) in **MS Word 2019 on Windows or Word 2016 and later on Mac**:

1. Insert a picture in the document.
2. Right-click the image and select “Edit Alt Text”.
3. In the “alt text” section, provide your text description of the image.

Below are the steps to insert figure descriptions in **MS Word 2013 and 2016**:

1. Insert a picture in the document.
2. Right click on the inserted picture and select the **Format Picture** option.
3. In the settings at the right side of the window, click on the “Layout & Properties” icon (3rd option).
4. Expand **Alt Txt** option.
5. In the “Title” and “Description” text boxes, type the text you want to represent the figure, and then click “Close.”

Below are steps to insert the alt-txt value in **MS Word 2010/2011 for Windows\***:

1. Insert a picture in the document.
2. Right click on the inserted picture and select the **Format Picture** option.
3. Select the **Alt Txt** option from the left-side panel options.
4. In the “Title” and “Description” text boxes, type the text you want to represent the picture, and then click “Close.”  
   \* The Mac 2011 version 14.0.0 and later allows the option for inserting “alt-text.” In the MAC version of Word 2016, right-click on the image and select “Edit Alt Text” from the pop-up menu and then enter the description for the alt text.
   1. Quotations and Extracts

There are styles for block quotations, which should be used for quotes that are separated from in-line text. Below is an example.

“Microsoft tried to revive the idea of an assistant with Clippy, who began popping up in Microsoft Office in 1997. Its creator, Kevan Atteberry, was actually contracted by Microsoft to design Clippy, which, funnily enough, he did on a Mac … Sure, people could disable Clippy, but the fact he was on by default angered people.” [10]

* 1. Equations

There are two types of math equations: the *numbered display math equation* and the *un-numbered display math equation*. Below are examples of both.

* + 1. DisplayFormula.

The **DisplayFormula** style is applied in the numbered math equation. A numbered display equation always has an equation number (label) on the right.

(1)

* + 1. DisplayFormula.Unnum.

The **DisplayFormulaUnnum** style is applied only in unnumbered equations. An unnumbered display equation never contains an equation number Bertot and Grimes (2012) on the right—this element distinguishes it from the numbered equation.

Please note: the subsequent text after the **DisplayFormula** (numbered equation) or **DisplayFormulaUnnum** (unnumbered equation) must have the paragraph style **ParaContinue** applied.

* 1. Math statements

Math statements should have the “Statement” style applied.

**Theorem/Proof/Lemma.** Math statements should have the “**Statement**” style applied. This paragraph is an example of the “**Statement**” style.

* 1. Algorithms

Algorithms use the styles “AlgorithmCaption” and “Algorithm”.

ALGORITHM 1: Iterative Algorithm

current\_position center

current\_direction up

current\_position is inside circle

while current\_position is inside circle, do

neighborhood all grid hexes within two hexes from current\_position

for each hex in neighborhood, do

for each neuron in hex do

convert neuron\_orientation to vector

scale vector by neuron\_excitation

vector\_sum vector\_sum + vector

end

end

normalize vector\_sum

end

1. COMPUTER CODE

import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.Scanner;  
  
class Course {  
 private String courseCode;  
 private String courseName;  
 private String instructor;  
 private String timeSlot;  
 private String room;  
  
 public Course(String courseCode, String courseName, String instructor, String timeSlot, String room) {  
 this.courseCode = courseCode;  
 this.courseName = courseName;  
 this.instructor = instructor;  
 this.timeSlot = timeSlot;  
 this.room = room;  
 }  
  
 public String getCourseCode() {  
 return courseCode;  
 }  
  
 public String getCourseName() {  
 return courseName;  
 }  
  
 public String getInstructor() {  
 return instructor;  
 }  
  
 public String getTimeSlot() {  
 return timeSlot;  
 }  
  
 public String getRoom() {  
 return room;  
 }  
  
 @Override  
 public String toString() {  
 return courseCode + " - " + courseName + " | Instructor: " + instructor + " | Time: " + timeSlot + " | Room: " + room;  
 }  
}  
  
class Student {  
 private String studentID;  
 private String name;  
 private Map<String, Integer> grades;  
  
 public Student(String studentID, String name) {  
 this.studentID = studentID;  
 this.name = name;  
 this.grades = new HashMap<>();  
 }  
  
 public String getStudentID() {  
 return studentID;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public Map<String, Integer> getGrades() {  
 return grades;  
 }  
  
 public void addGrade(String courseCode, int grade) {  
 grades.put(courseCode, grade);  
 }  
}  
  
public class CollegeStudentSchedulingSyste {  
 public static void main(String[] args) {  
 List<Course> courses = new ArrayList<>();  
 List<Student> students = new ArrayList<>();  
 Scanner scanner = new Scanner(System.*in*);  
  
 while (true) {  
 System.*out*.println("College Student Scheduling System");  
 System.*out*.println("1. View Course Schedule");  
 System.*out*.println("2. Add Course");  
 System.*out*.println("3. Enroll Student");  
 System.*out*.println("4. Add Grade");  
 System.*out*.println("5. View Student Grades");  
 System.*out*.println("6. Exit");  
 System.*out*.print("Select an option: ");  
 int choice = scanner.nextInt();  
  
 switch (choice) {  
 case 1:  
 *viewCourseSchedule*(courses);  
 break;  
 case 2:  
 *addCourse*(courses);  
 break;  
 case 3:  
 *enrollStudent*(students);  
 break;  
 case 4:  
 *addGrade*(students, courses);  
 break;  
 case 5:  
 *viewStudentGrades*(students);  
 break;  
 case 6:  
 System.*out*.println("Exiting the system.");  
 scanner.close();  
 return;  
 default:  
 System.*out*.println("Invalid choice. Please try again.");  
 }  
 }  
 }  
  
 public static void viewCourseSchedule(List<Course> courses) {  
 System.*out*.println("Course Schedule:");  
 if (courses.isEmpty()) {  
 System.*out*.println("No courses registered.");  
 } else {  
 for (Course course : courses) {  
 System.*out*.println(course);  
 }  
 }  
 }  
  
 public static void addCourse(List<Course> courses) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter course code: ");  
 String courseCode = scanner.next();  
 System.*out*.print("Enter course name: ");  
 String courseName = scanner.next();  
 System.*out*.print("Enter instructor: ");  
 String instructor = scanner.next();  
 System.*out*.print("Enter time slot: ");  
 String timeSlot = scanner.next();  
 System.*out*.print("Enter room: ");  
 String room = scanner.next();  
  
 Course course = new Course(courseCode, courseName, instructor, timeSlot, room);  
 courses.add(course);  
 System.*out*.println("Course added to the schedule.");  
 }  
  
 public static void enrollStudent(List<Student> students) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter student ID: ");  
 String studentID = scanner.next();  
 System.*out*.print("Enter student name: ");  
 String studentName = scanner.next();  
  
 Student student = new Student(studentID, studentName);  
 students.add(student);  
 System.*out*.println("Student enrolled.");  
 }  
  
 public static void addGrade(List<Student> students, List<Course> courses) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter student ID: ");  
 String studentID = scanner.next();  
 System.*out*.print("Enter course code: ");  
 String courseCode = scanner.next();  
 System.*out*.print("Enter grade: ");  
 int grade = scanner.nextInt();  
  
 Student student = *findStudentById*(students, studentID);  
 if (student != null) {  
 if (*isCourseInSchedule*(courses, courseCode)) {  
 student.addGrade(courseCode, grade);  
 System.*out*.println("Grade added for " + student.getName() + " in " + courseCode);  
 } else {  
 System.*out*.println("Course not found in the schedule.");  
 }  
 } else {  
 System.*out*.println("Student not found.");  
 }  
 }  
  
 public static Student findStudentById(List<Student> students, String studentID) {  
 for (Student student : students) {  
 if (student.getStudentID().equals(studentID)) {  
 return student;  
 }  
 }  
 return null;  
 }  
  
 public static boolean isCourseInSchedule(List<Course> courses, String courseCode) {  
 for (Course course : courses) {  
 if (course.getCourseCode().equals(courseCode)) {  
 return true;  
 }  
 }  
 return false;  
 }  
  
 public static void viewStudentGrades(List<Student> students) {  
 Scanner scanner = new Scanner(System.*in*);  
 System.*out*.print("Enter student ID: ");  
 String studentID = scanner.next();  
  
 Student student = *findStudentById*(students, studentID);  
 if (student != null) {  
 System.*out*.println("Grades for " + student.getName() + ":");  
 Map<String, Integer> grades = student.getGrades();  
 if (grades.isEmpty()) {  
 System.*out*.println("No grades recorded.");  
 } else {  
 for (Map.Entry<String, Integer> entry : grades.entrySet()) {  
 System.*out*.println(entry.getKey() + ": " + entry.getValue());  
 }  
 }  
 } else {  
 System.*out*.println("Student not found.");  
 }  
 }  
}

1. CITING RELATED WORK

Bardadym, V. A. (1995, August). Computer-aided school and university timetabling: The new wave. In international conference on the practice and theory of automated timetabling (pp. 22-45). Berlin, Heidelberg: Springer Berlin Heidelberg[1].

Reis, L. P., & Oliveira, E. (2001). A language for specifying complete timetabling problems. In Practice and Theory of Automated Timetabling III: Third International Conference, PATAT 2000 Konstanz, Germany, August 16–18, 2000 Selected Papers 3 (pp. 322-341). Springer Berlin Heidelberg[2].

McCollum, McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A. J., ... & Burke, E. K. (2010). Setting the research agenda in automated timetabling: The second international timetabling competition. Informs Journal on computing, 22(1), 120-130[3].

Chelsea Burke, E., & Rudová, H. (Eds.). (2007). Practice and Theory of Automated Timetabling VI: 6th International Conference, PATAT 2006 Brno, Czech Republic, August 30-September 1, 2006 Revised Selected Papers (Vol. 3867). Springer[4].

Jon M. Cooper, T. B., & Kingston, J. H. (1993). The solution of real instances of the timetabling problem. The computer journal, 36(7), 645-653[5].

Burke, E. K., Elliman, D. G., & Weare, R. (1994). A university timetabling system based on graph colouring and constraint manipulation. Journal of research on computing in education, 27(1), 1-18[6]

ACKNOWLEDGMENTS

Acknowledgments are placed before the references. Add information about grants, awards, or other types of funding that you have received to support your research. Author can capture the **grant sponsor information**, by selecting the grant sponsor text and apply style ‘GrantSponsor’. After this, select grant no and apply ‘GrantNumber’ from style panel. Example of Grant sponsor: Competitive Research Programme and example of Grant no: CRP 10-2012-03.

REFERENCES

1. Anisimov B.V. and Vlasov V.P. (1971) “Analysis of Decision Rules Applied to Lessons Timetabling”, *Mathematical and Information Problems of Science Prognostication and Management*, Kiev, V. Glushkov Inst. for Cybernetics, 113–122 (Russian).
2. Y.H. Cho et al.[A personalized recommender system based on web usage mining and decision tree induction](https://www.sciencedirect.com/science/article/pii/S0957417402000520)Expert Systems with Applications(2002)
3. Barnes, T., H. Richter, E. Powell, A. Chaffin, A. Godwin. Game2Learn: Building CS1 learning games for retention. ITiCSE 2007, Dundee, Scotland, June 25--27, 2007.
4. K. Solanki and P. Pittalia, "Review of face recognition techniques", Int. J. Comput. Appl., vol. 133, pp. 20-24, Jan. 2016.
5. Soni, A., Kumar, V., Kaur, R., and Hemavath, D., Predicting student performance using data mining techniques. *International Journal of Pure and applied Mathematics* 119(12):221–227, 2018.
6. Amjad Altadmri and Neil C.C. Brown. 2015. 37 Million Compilations: Investigating Novice Programming Mistakes in Large-Scale Student Data. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education (SIGCSE '15). ACM, New York, NY, USA, 522--527. x978--1--4503--2966--8 https://doi.org/10.1145/2676723.2677258)

  APPENDICES

In the appendix section, three levels of Appendix headings are available.

A.1 General Guidelines (AppendixH2)

1. Save as you go and backup your file regularly.
2. Do not work on files that are saved in a cloud directory. To avoid problems such as MS Word crashing, please only work on files that are saved locally on your machine.
3. Equations should be created with the built-in Microsoft® Equation Editor included with your version of Word. (Please check the compatibility at <http://tinyurl.com/lzny753> for using MathType.)
4. Please save all files in DOCX format, as the DOC format is only supported for the Mac 2011 version.
5. Tables should be created with Word’s “Insert Table” tool and placed within your document. (Tables created with spaces or tabs will have problems being properly typeset. To ensure your table is published correctly, Word’s table tool must be used.)
6. Do not copy-and-paste elements into the submission document from Excel such as charts and tables.
7. Footnotes should be inserted using Word’s “Insert Footnote” feature.
8. Do not use Word’s “Insert Shape” function to create diagrams, etc.
9. Do not have references appear in a table/cells format as it will produce an error during the layout generation process.
10. MS Word does not consistently allow the original formatting to be modified in the text. In these cases, it is best to copy all the document’s text from the specific file and paste into a new MS Word document and then save it.
11. At times there are font problems such as “odd” stuff/junk characters that appear in the text, usually in the references. This can be caused by a variety of reasons such as copying-and-pasting from another file, file transfers, etc. Please review your text prior to submission to make sure it reads correctly.

A.1.1 Preparing Graphics (AppendixH3)

1. Accepted image file formats: TIFF (.tif), JPEG (.jpg).
2. Scalable vector formats (i.e., SVG, EPS and PS) are greatly preferred.
3. Application files (e.g., Corel Draw, MS Word, MS Excel, PPT, etc.) are NOT recommended.
4. Images created in Microsoft Word using text-box, shapes, clip-art are NOT recommended.
5. IMPORTANT: All fonts must be embedded in your figure files.
6. Set the correct orientation for each graphics file.

A.2 Placeholder Text

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Vulputate sapien nec sagittis aliquam. Malesuada fames ac turpis egestas sed tempus urna. Posuere sollicitudin aliquam ultrices sagittis orci. Consequat id porta nibh venenatis cras sed felis eget. Pellentesque eu tincidunt tortor aliquam nulla facilisi cras fermentum odio. Tincidunt nunc pulvinar sapien et ligula ullamcorper malesuada proin. Tincidunt lobortis feugiat vivamus at augue. Eget nunc lobortis mattis aliquam faucibus. Egestas diam in arcu cursus euismod quis.

Erat pellentesque adipiscing commodo elit at imperdiet. In hac habitasse platea dictumst quisque sagittis purus. At lectus urna duis convallis. Eu mi bibendum neque egestas congue. Est ullamcorper eget nulla facilisi etiam dignissim diam. Sed ullamcorper morbi tincidunt ornare massa eget. Aenean vel elit scelerisque mauris pellentesque. Ullamcorper dignissim cras tincidunt lobortis feugiat vivamus. Cras fermentum odio eu feugiat pretium nibh. Congue eu consequat ac felis donec et odio pellentesque diam. Velit sed ullamcorper morbi tincidunt ornare massa eget egestas. In metus vulputate eu scelerisque felis imperdiet proin fermentum leo. Nulla malesuada pellentesque elit eget gravida cum.

Nullam ac tortor vitae purus faucibus ornare suspendisse. Libero enim sed faucibus turpis in eu mi bibendum neque. Sodales ut etiam sit amet nisl purus. Egestas diam in arcu cursus. Aliquet porttitor lacus luctus accumsan tortor. Pharetra magna ac placerat vestibulum lectus. Sit amet mauris commodo quis imperdiet massa tincidunt. In nisl nisi scelerisque eu ultrices vitae auctor. Nisi vitae suscipit tellus mauris a diam. Dui vivamus arcu felis bibendum ut tristique. Laoreet suspendisse interdum consectetur libero id.

Enim eu turpis egestas pretium. Nulla aliquet enim tortor at auctor urna. Id aliquet risus feugiat in. Non enim praesent elementum facilisis leo. Integer feugiat scelerisque varius morbi enim nunc faucibus. Egestas dui id ornare arcu odio ut sem nulla pharetra. Montes nascetur ridiculus mus mauris. Orci dapibus ultrices in iaculis. Enim sed faucibus turpis in eu mi bibendum neque. Faucibus pulvinar elementum integer enim neque volutpat ac tincidunt vitae. Et ultrices neque ornare aenean euismod elementum. Et pharetra pharetra massa massa ultricies mi quis hendrerit dolor. Tempus iaculis urna id volutpat lacus laoreet non curabitur gravida. Est velit egestas dui id ornare arcu odio. Eu facilisis sed odio morbi quis commodo odio. Lectus vestibulum mattis ullamcorper velit sed ullamcorper morbi tincidunt.

Eu non diam phasellus vestibulum lorem sed risus ultricies. Convallis aenean et tortor at risus viverra adipiscing at. Mauris pellentesque pulvinar pellentesque habitant morbi. Elementum sagittis vitae et leo duis. Massa enim nec dui nunc. Nisl tincidunt eget nullam non nisi est sit amet. Amet nisl purus in mollis nunc sed id semper. Fermentum leo vel orci porta non pulvinar neque laoreet suspendisse. Diam vel quam elementum pulvinar etiam non quam. Sagittis orci a scelerisque purus semper eget. Aliquet porttitor lacus luctus accumsan tortor. Integer vitae justo eget magna fermentum iaculis eu non diam. Egestas pretium aenean pharetra magna ac. Cursus metus aliquam eleifend mi in nulla. Cursus mattis molestie a iaculis at erat pellentesque adipiscing. Pulvinar pellentesque habitant morbi tristique senectus. Gravida cum sociis natoque penatibus et magnis dis parturient montes. In aliquam sem fringilla ut. Ut consequat semper viverra nam libero justo laoreet. Pellentesque diam volutpat commodo sed egestas.

Ornare arcu odio ut sem nulla pharetra diam. Ut enim blandit volutpat maecenas volutpat blandit aliquam. Tempus iaculis urna id volutpat lacus. Nascetur ridiculus mus mauris vitae. Venenatis cras sed felis eget velit aliquet sagittis id. Laoreet non curabitur gravida arcu ac tortor dignissim convallis aenean. Maecenas ultricies mi eget mauris pharetra et ultrices neque ornare. Egestas purus viverra accumsan in nisl nisi scelerisque eu ultrices. Tempus urna et pharetra pharetra massa massa. Pulvinar neque laoreet suspendisse interdum consectetur libero id.

Nisl rhoncus mattis rhoncus urna neque viverra justo nec ultrices. Morbi quis commodo odio aenean sed adipiscing diam donec. Neque gravida in fermentum et. Scelerisque purus semper eget duis at tellus. Volutpat blandit aliquam etiam erat velit scelerisque in dictum non. Odio ut sem nulla pharetra diam sit. Sed pulvinar proin gravida hendrerit lectus a. Diam ut venenatis tellus in metus vulputate eu scelerisque. Id semper risus in hendrerit. Vel quam elementum pulvinar etiam. Amet aliquam id diam maecenas ultricies mi. Auctor elit sed vulputate mi sit amet. Orci dapibus ultrices in iaculis nunc. Sed vulputate odio ut enim blandit volutpat maecenas volutpat. Auctor urna nunc id cursus metus. Integer enim neque volutpat ac tincidunt vitae.

Scelerisque in dictum non consectetur a erat. Vel risus commodo viverra maecenas accumsan lacus vel facilisis volutpat. Dignissim sodales ut eu sem integer vitae justo eget magna. Nunc non blandit massa enim nec dui nunc mattis enim. Sed vulputate odio ut enim blandit volutpat maecenas. Ante in nibh mauris cursus. Donec pretium vulputate sapien nec sagittis aliquam malesuada. Eu volutpat odio facilisis mauris sit amet massa. Blandit turpis cursus in hac habitasse platea dictumst quisque. Donec enim diam vulputate ut pharetra sit.

Magna fringilla urna porttitor rhoncus dolor purus non. Fames ac turpis egestas integer eget. Mattis rhoncus urna neque viverra. Laoreet sit amet cursus sit amet dictum sit amet. Vel pretium lectus quam id leo in vitae turpis massa. Euismod lacinia at quis risus sed vulputate odio ut. Lorem dolor sed viverra ipsum. Viverra justo nec ultrices dui sapien. Aliquam nulla facilisi cras fermentum odio eu feugiat pretium. Adipiscing commodo elit at imperdiet dui accumsan sit amet nulla. Morbi leo urna molestie at elementum eu facilisis sed. Habitant morbi tristique senectus et netus et malesuada. Viverra ipsum nunc aliquet bibendum enim. Integer vitae justo eget magna fermentum. Tincidunt id aliquet risus feugiat. Mauris ultrices eros in cursus turpis. Amet venenatis urna cursus eget nunc. Nisl nisi scelerisque eu ultrices vitae.

Non pulvinar neque laoreet suspendisse interdum consectetur libero. Facilisis leo vel fringilla est ullamcorper eget nulla facilisi. Ipsum dolor sit amet consectetur adipiscing elit pellentesque. Risus quis varius quam quisque id. Bibendum arcu vitae elementum curabitur vitae. Vitae et leo duis ut diam quam nulla. Orci eu lobortis elementum nibh tellus molestie nunc non blandit. Arcu odio ut sem nulla pharetra diam sit amet. Quis vel eros donec ac odio. Est lorem ipsum dolor sit amet consectetur adipiscing

1. \* Place the footnote text for the author (if applicable) here. [↑](#footnote-ref-1)