COLORADO STATE UNIVERSITY

GLOBAL

CSC505: PRINCIPLES OF SOFTWARE DEVELOPMENT

Credit Hours: 3

Contact Hours: This is a 3-credit course, offered in accelerated format. This means that 16 weeks of material is covered in 8 weeks. The exact number of hours per week that you can expect to spend on each course will vary based upon the weekly coursework, as well as your study style and preferences. You should plan to spend 14-20 hours per week in each course reading material, interacting on the discussion boards, writing papers, completing projects, and doing research.

Faculty Information: Faculty contact information and office hours can be found on the faculty profile page.

COURSE DESCRIPTION AND OUTCOMES

Course Description:

This course provides students with an integrated and detailed approach to programming and software development principles. Students will gain an understanding of object-oriented software programming and pertinent software development principles. Topics included in this course focus on core programming concepts, data structures, methods, classes, and software models.

Course Overview:

This course deals with the creation of cost-effective, practical, and robust computing and IT solutions while addressing problems in these domains using modern techniques. In particular, students will gain understanding of relevant topics in object-oriented software programming as well as software development principles using UML and related tools.

Course Learning Outcomes:

1. Using a software model, create an algorithm to solve a specific software problem.
2. Using basic data types in a software application, identify conditional and repetitive data structures in programming.
3. Implement an application using appropriate data structures to solve a given programming problem.
4. Compare software development models.
5. Select an appropriate software development model to solve a specific software problem.

PARTICIPATION & ATTENDANCE

Prompt and consistent attendance in your online courses is essential for your success at CSU Global Campus. Failure to verify your attendance within the first 7 days of this course may result in your withdrawal. If for some reason you would like to drop a course, please contact your advisor.

Online classes have deadlines, assignments, and participation requirements just like on-campus classes. Budget your time carefully and keep an open line of communication with your instructor. If you are having technical problems, problems with your assignments, or other problems that are impeding your progress, let your instructor know as soon as possible.

Course Materials

Required:

Pressman, R. & Maxim, B. (2020). *Software engineering: A practitioner's approach* (9th ed.). New York, NY: McGraw Hill. ISBN-13: 9781259872976

Python 3 and related IDEs

Suggested:

Sommerville, I. (2019). *Engineering software products: An introduction to modern software engineering.* New York, NY: Pearson. ISBN-13: 9780135210642

***NOTE:** All non-textbook required readings and materials necessary to complete assignments, discussions, and/or supplemental or required exercises are provided within the course itself. Please read through each course module carefully.*

Course Schedule

Due Dates

The Academic Week at CSU Global begins on Monday and ends the following Sunday.

- **Discussion Boards:** The original post must be completed by Thursday at 11:59 p.m. MT and Peer Responses posted by Sunday 11:59 p.m. MT. Late posts may not be awarded points.
- **Critical Thinking:** Assignments are due Sunday at 11:59 p.m. MT.

Weekly Reading and Assignment Details

Module 1

Readings

- Chapter 1 in *Software Engineering: A Practitioner's Approach*
- Ju, A. & Fox, A. (2018, July). TEAMSCOPE: Measuring software engineering processes with teamwork telemetry. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018).* Association for Computing Machinery. New York, NY, USA, 123–128.
- Weinman, B. (2018). *Installing Python for Windows*.
- Weinman, B. (2018). *Installing Python for Mac*.

Discussion (25 points)

Critical Thinking (50 points)

OPTION #1: Standard Software Systems Development Questions
For this assignment, you will first write a Python script and then answer some standard software systems development questions about the script you have written. Programmers, whether in teams or as individuals, have to be mindful of challenges during computer-based systems development. Put yourself in the position of a software developer.

Complete the following:

1. First, in order to get a feel for Python programming and related IDE, write a Python script.
2. Next, answer the following questions in a separate Word document. Write out both the question and your answer:
   - Why does it take so long to get software finished?
   - Why are development costs so high?
   - Why can't we find all errors before we give the software to our customers?
   - Why do we spend so much time and effort maintaining existing programs?
   - Why do we continue to have difficulty in measuring progress, as software is being developed and maintained?

Your questions and answers should be at least 1 page in length and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Submit the source file (.py), the screenshots of a successful execution of your program, all related diagrams, and the questions and answers in a zipped folder.

OPTION #2: Umbrella Activity
Umbrella activities are common in the software development process. For this assignment, describe the nature of these umbrella activities and then develop a UML activity diagram to model them. If you are unfamiliar with UML activity diagrams, refer to the definition found at *UML Composition* (link below) or use other resources to learn more about them. Then, using Python, implement an activity class to print out the contents of your UML diagram.

Complete the following:

- Write a paper describing umbrella activities. Include a UML activity diagram.
- Complete a Python activity class based on the diagram.

Your paper should be 1-2 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Submit the source file (.py), the screenshots of a successful execution of your program, all related diagrams, and the paper in a zipped folder.

Links:
UML Composition
Drawing UML With Plant UML

Module 2

Readings

- Chapter 2 in *Software Engineering: A Practitioner's Approach*
- Lenarduzzi, V. Sillitti, A. & Taibi, D. (2017). Analyzing forty years of software maintenance models. In *Proceedings of the 39th International Conference on Software Engineering Companion (ICSE-C '17).* IEEE Press, 146–148.
- Baltes, S. & Diehl, S. (2018). Towards a theory of software development expertise. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2018).* Association for Computing Machinery. New York, NY, USA, 187–200.
- Gallina, B., Muram, F., & Castellanos Ardila, J. (2018). Compliance of agilized (software) development processes with safety standards: A vision. In *Proceedings of the 19th International Conference on Agile Software Development: Companion (XP '18).* Association for Computing Machinery. New York, NY, USA, Article 14, 1–6.

Discussion (25 points)

Critical Thinking (80 points)

OPTION #1: Mutually Conflicting Requirements

As a software developer seeking specifications from users, it is possible to encounter two stakeholders who have conflicting ideas about the specifications. In particular, they may disagree on what the software should be. This phenomenon is known as "mutually conflicting requirements."

For this assignment, complete the following:

- Develop a process pattern that addresses this challenge, and then use UML to create a communication diagram for stakeholders that prompts a user to input his specifications and then returns the object of the user's input.
- Use Python to implement your communication diagram.

Use one of the following UML tools (links below) to create your diagrams for this assignment:

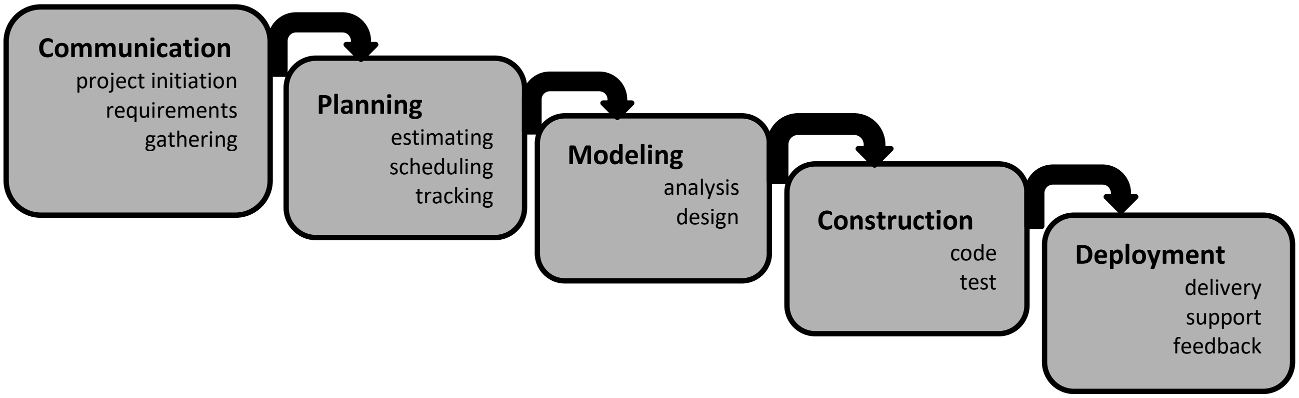
- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
UMLet
Gliffy

OPTION #2: Redesigning the Waterfall Model

Consider the following Waterfall Model. Study it and make a list of some of its shortcomings. Then, for this assignment, use your list of shortcomings to modify the address the shortcomings. Title it with your last name and the word Model (YourLastName Model).



Complete the following:

- Using the UML diagram of your choice, draft YourLastName Diagram making sure you clearly indicate your additions.
- Using Python, implement the YourLastName class that prompts the user to the key elements of the diagram and returns these objects in a well-formatted output.

Use one of the following UML tools (links below) to create your diagrams for this assignment:

- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
UMLet
Gliffy

Module 3

Readings

- Chapter 3 & 4 in *Software Engineering: A Practitioner's Approach*
- Scrum.org. (2020). *A better way of building products*. Retrieved from https://www.scrum.org/resources/what-is-scrum.
- Ardito, C., Baldassarre, M. T., Caivano, D. & Lanzilotti, R. (2017). Integrating a SCRUM-based process with human centred design: an experience from an action research study. In *Proceedings of the 5th International Workshop on Conducting Empirical Studies in Industry (CESI '17).* IEEE Press, 2–8.
- Delgado Pinto, T., Gonçalves, W. I. &and Veiga Costa, P. (2019). User interface prototype generation from agile requirements specifications written in concordia. In *Proceedings of the 25th Brazilian Symposium on Multimedia and the Web (WebMedia '19*). Association for Computing Machinery. New York, NY, USA, 61–64.

Discussion (25 points)

Critical Thinking (80 points)

OPTION #1: Mobile App Prototype

You have been contacted and are thinking about developing a "first" prototype for a mobile app that let users save a shopping list on their devices. How would you create a preliminary architectural design for it?

For this assignment, complete the following:

- Choose the necessary tools (UML or other diagramming tools) to show your strategy.
- Create a series of sketches representing the key screens for a paper prototype for the shopping list app.
- Using Python, write a script that will print out (1) pseudocode of your strategy and (2) a list of tools used to develop the prototype.

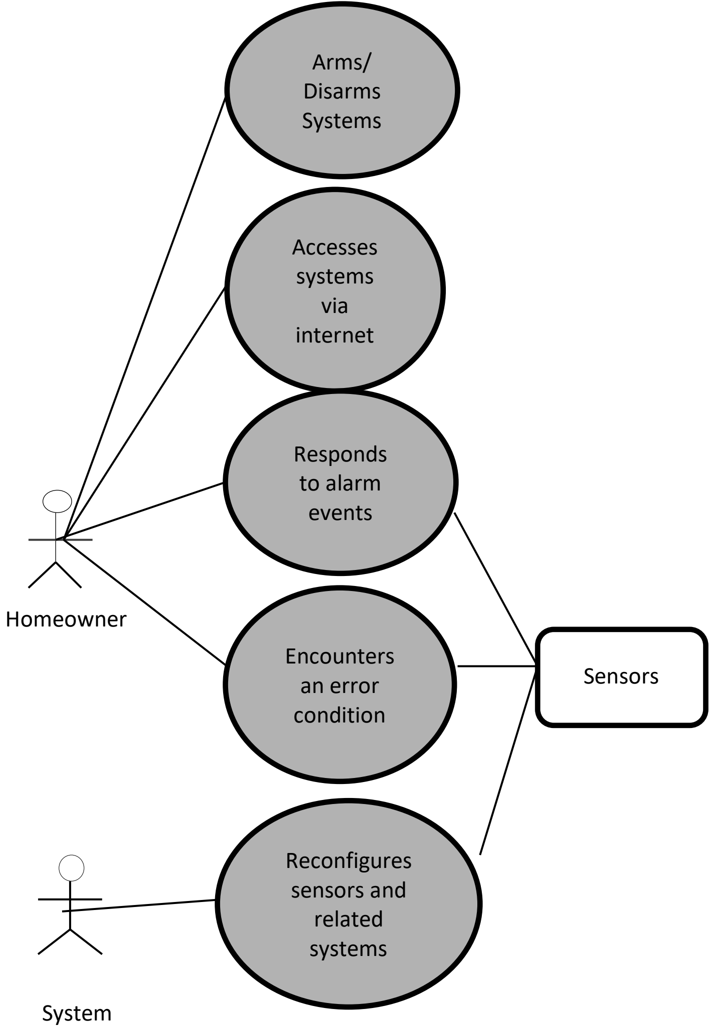
Feel free to use one of the following UML tools (links below) to create your diagrams for this assignment:

- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
UMLet
Gliffy

OPTION #2: Agile Mentality: User Story
After reading the Agile Manifesto, write a user story that describes the “favorite sports” or “favorites” feature available on the user’s phone.


Arms/ Disarms Systems
Accesses systems via internet
Responds to alarm events
Encounters an error condition
Reconfigures sensors and related systems
Homeowner
System
Sensors

(Adapted from Pressman, 2020, p. 118)

Using the SafeHome UML diagram as a template (you can also a different UML Diagram type), create a user story and implement it in Python.

Complete the following:

- A user story UML diagram that describes the “favorite sports” or “favorites” feature available on the user’s phone.

- A Python implementation of your design.

Use one of the following UML tools (links below) to create your diagrams for this assignment:

- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
Comprehensive Guide to the Agile Manifesto
UMLet
Gliffy

Module 4

Readings

- Chapter 5 in *Software Engineering: A Practitioner's Approach*
- Marsicano, G., Pereira, D.V., da Silva, F. Q. B., & França, C. (2017). Team maturity in software engineering teams. In *Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement* (ESEM '17). IEEE Press, 235–240.
- Pereira, D.V., Marsicano Corrêa, G., da Silva, F. B. Q., & Monteiro Ribeiro, D. (2017). Team maturity in software engineering teams: A work in progress. In *Proceedings of the 10th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '17).* IEEE Press, 70–73.

Discussion (25 points)

Critical Thinking (80 points)

OPTION #1: Common Personality Traits

Based on your personal experience, what makes an excellent software developer? For this assignment, select three personality traits common among excellent programmers and then implement it as a Python script.

Complete the following:

- Using a UML diagram of your choice, depict three personality traits that appear to be common among excellent software developers.
- Implement the results of the diagramming activity as a Python script. Use OODesign.com's Builder Pattern (link below) for inspiration.

Use one of the following UML tools (links below) to create your diagrams for this assignment:

- UMLet
- Gliffy

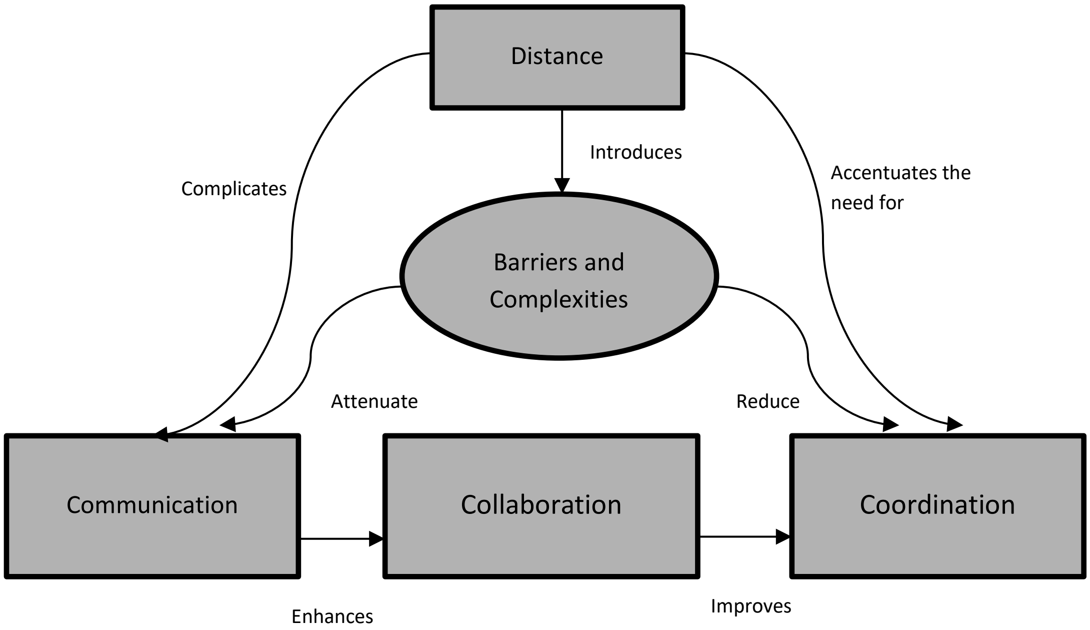
Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
https://www.oodesign.com/builder-pattern.html

UMLet
Gliffy

OPTION #2: Builder Design Pattern

Consider the following UML diagram that represents factors affecting the GSD Software team.



(Adapted from Pressman, 2020, p. 118)

Using this diagram as a guide, imagine a situation in which the CSC505 course software team members make use of one or more forms of social media as part of their software project.

Complete the following:

- Create a diagram to depict how the CSC505 course software team members use social media as part of a software project.
- Implement the results as a Python script.

Use one of the following UML tools (links below) to create your diagrams for this assignment:

- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
UMLet
Gliffy

Portfolio Milestone (25 points)

In your first three Critical Thinking Assignments, chances are you missed a concept and did not earn all the possible points. This is an opportunity to correct any mistakes you may have made. Go ahead and make appropriate corrections to the code/assignment you submitted in the Critical Thinking

Assignments for Modules 1 through 3. Corrections should reflect feedback from your instructor and improvements in execution, organization, and style. Resubmit your programs with all outlined corrections. If no corrections are necessary, resubmit your original assignments.

Module 5

Readings

- Chapters 7 & 8 in *Software Engineering: A Practitioner's Approach*
- Jantunen, A., Dumdum, R. & Gause, D. C. (2019). Towards new requirements engineering competencies. In *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '19).* IEEE Press, 131–134.
- Chanin, R., Pompermaier, L., Sales, A. & Prikladnicki, R. (2019). Collaborative practices for software requirements gathering in software startups. In *Proceedings of the 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE '19).* IEEE Press, 31–32.
- Meriah, I. Rabai, L. B. A. (2018). A survey of quantitative security risk analysis models for computer systems. In *Proceedings of the 2nd International Conference on Advances in Artificial Intelligence (ICAAI 2018).* Association for Computing Machinery*.* New York, NY, USA, 36–40.

Discussion (25 points)

Critical Thinking (80 points)

OPTION #1: Web-based Pothole Tracking and Repair System
The department of public works for a large city has decided to develop a web-based pothole tracking and repair system (PHTRS). A description follows:

> Citizens can log onto a website and report the location and severity of potholes. As potholes are reported, they are logged within a "public works department repair system" and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.), district (determined from street address), and repair priority (determined from the size of the pothole).
>
> Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used).
>
> Finally, a damage file is created to hold information about reported damage due to the pothole and includes citizen's name, address, phone number, type of damage, and dollar amount of damage. PHTRS is an online system; all queries are to be made interactively (Pressman & Maxim, 2010, p. 155).

Complete the following:

- Create a UML use case diagram for a PHTRS system. You'll have to make a number of assumptions about the manner in which a user interacts with this system.
- Implement your system using Python.

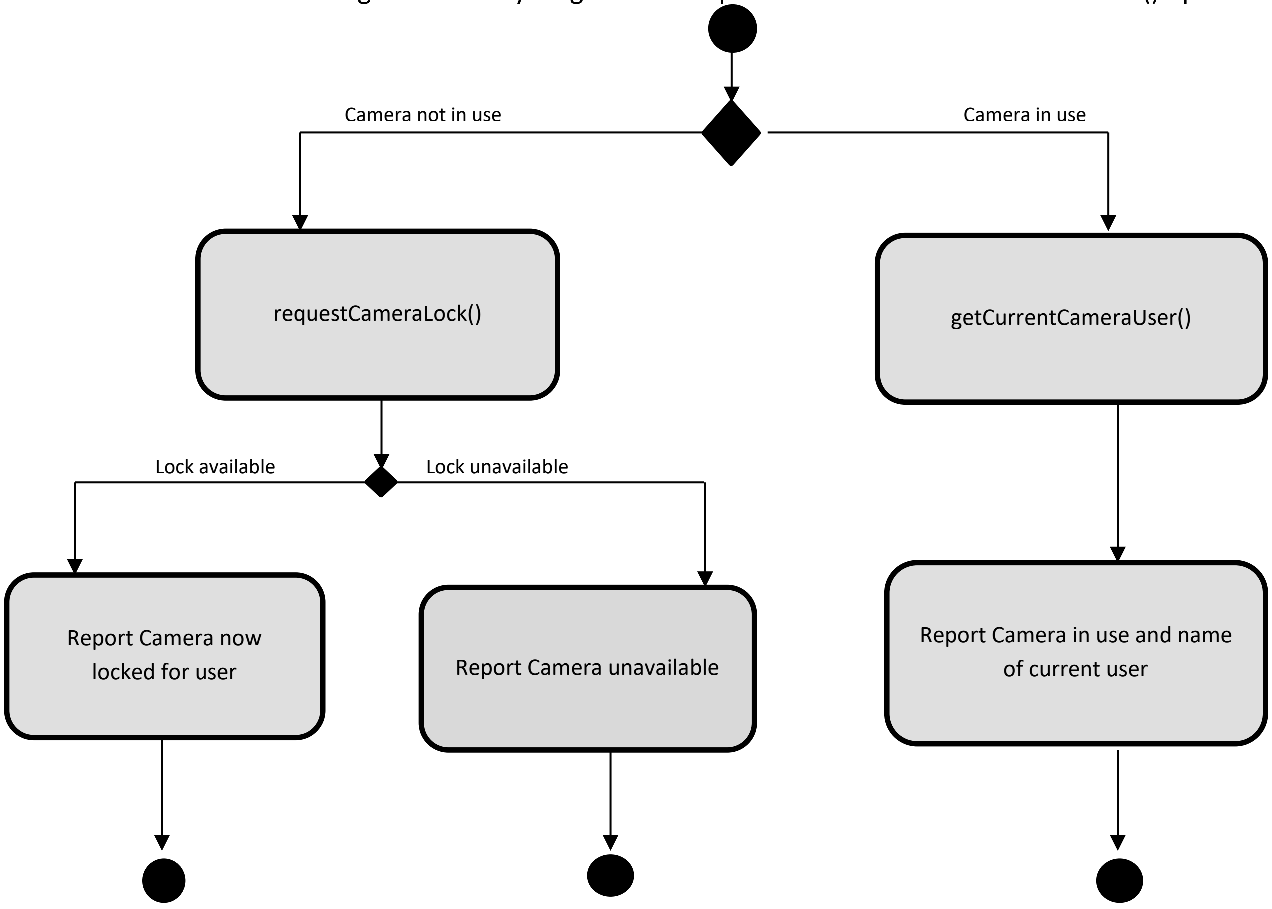
Use one of the following UML tools (links below) to create your diagrams for this assignment:

- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
UMLet
Gliffy

OPTION #2: One Aspect of PHTRS
Consider the following UML activity diagram that represents the takeControlOf Camera() operation.


Camera not in use
Camera in use
requestCameraLock()
getCurrentCameraUser()
Lock available
Lock unavailable
Report Camera now locked for user
Report Camera unavailable
Report Camera in use and name of current user

(Adapted from Pressman, 2020, p. 147)

Then, review this information:

- The department of public works for a large city has decided to develop a web-based pothole tracking and repair system (PHTRS). A description follows:

  > Citizens can log onto a website and report the location and severity of potholes. As potholes are reported, they are logged within a “public works department repair system” and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.), district (determined from street address), and repair priority (determined from the size of the pothole).
  >
  > Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not

> repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used).

For this assignment, complete the following:

- Develop an activity diagram for one aspect of PHTRS by using the diagram template as a guide and the information provided on PHTRS.
- Implement your system using Python.

Use one of the following UML tools (links below) to create your diagrams for this assignment:

- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
UMLet
Gliffy

Module 6

Readings

- Chapters 9 & 10 in *Software Engineering: A Practitioner's Approach*
- Kun, P., Mulder, I., de Götzen, A. & Kortuem, G. (2019). Creative data work in the design process. In *Proceedings of the 2019 on Creativity and Cognition (C&C '19).* Association for Computing Machinery. New York, NY, USA, 346–358.
- Gonçales, L. J., Farias, K., & Bischoff, V. 2019. On the effects of developers' intuition on measuring similarity between UML models. In *Proceedings of the XV Brazilian Symposium on Information Systems (SBSI'19).* Association for Computing Machinery. New York, NY, USA, Article 29, 1–8.
- Bhat, M., Shumaiev, K. & Matthes, F. (2017). Towards a framework for managing architectural design decisions. In *Proceedings of the 11th European Conference on Software Architecture: Companion Proceedings (ECSA '17).* Association for Computing Machinery, New York, NY, USA, 48–51.

Discussion (25 points)

Critical Thinking (80 points)

OPTION #1: Web-based Pothole Tracking and Repair System, Part 2
The department of public works for a large city has decided to develop a web-based pothole tracking and repair system (PHTRS). A description follows:

> Citizens can log onto a website and report the location and severity of potholes. As potholes are reported they are logged within a "public works department repair system" and are assigned an identifying number, stored by street address, size (on a scale of 1 to 10), location (middle, curb, etc.), district (determined from street address), and repair priority (determined from the size of the pothole). Work order data are associated with each pothole and include pothole location and size, repair crew identifying number, number of people on crew, equipment assigned, hours applied to repair, hole status (work in progress, repaired, temporary repair, not repaired), amount of filler material used, and cost of repair (computed from hours applied, number of people, material and equipment used). Finally, a damage file is created to hold information

> about reported damage due to the pothole and includes citizen’s name, address, phone number, type of damage, and dollar amount of damage. PHTRS is an online system; all queries are to be made interactively (Pressman & Maxim, 2010, p. 155).

In Module 5, you encountered this problem and were asked to draw a UML use case diagram for the PHTRS system. You also had to make a number of assumptions about the manner in which a user interacts with this system and were asked to implement your system using Python.

For this assignment, complete the following:

- Use the Design Approach discussed in Chapter 10 of your textbook to design and implement the system.
- Use Python for your system implementation.

Use one of the following UML tools (links below) to create your diagrams for this assignment:

- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
UMLet
Gliffy

OPTION #2: Stepwise Refinement Approach

Apply a “stepwise refinement approach” to develop three different levels of procedural abstractions for one or more of the following programs.

For this assignment, complete the following:

1. Develop a check writer that, given a numeric dollar amount, will print the amount in words normally required on a check.
2. Iteratively solve for the roots of a transcendental equation.
3. Develop a simple task-scheduling algorithm for an operating system.

Use one of the following UML tools (links below) to create your diagrams for this assignment and implement your system in Python:

- UMLet
- Gliffy

Submit the source file (.py), the screenshots of a successful execution of your program, and all related diagrams in a zipped folder.

Links:
UMLet
Gliffy

Module 7

Readings

- Chapters 11 & 12 in *Software Engineering: A Practitioner's Approach*
- Kuznetsov, K., Avdiienko, V., Gorla, A. & Zeller, A. (2018). Analyzing the user interface of Android apps. In *Proceedings of the 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft '18).* Association for Computing Machinery, New York, NY, USA, 84–87.
- Murad, C. (2019). Tools to support voice user interface design. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19).* Association for Computing Machinery, New York, NY, USA, Article 72, 1–5.
- Sutika, T., Funilkul, S., Triyason, T. & Supattatham, M. (2018). Quality of smartphone user experience analysis: Focusing on smartphone screen brightness level for the elderly. In *Proceedings of the 10th International Conference on Advances in Information Technology (IAIT 2018).* Association for Computing Machinery, New York, NY, USA, Article 18, 1–6.

Discussion (25 points)

Portfolio Milestone (25 points)

In your first three Critical Thinking Assignments, chances are you missed a concept and did not earn all the possible points. This is an opportunity to correct any mistakes you may have made in the previous assignments. Make appropriate corrections to the code in the Module 4-6 assignments. Corrections should reflect feedback from your instructor and improvements in execution, organization, and style. Resubmit your programs from Modules 4-6 with all outlined corrections. If no corrections are necessary, resubmit your original assignments.

Module 8

Readings

- Chapters 15, 17, 18 & 19 in *Software Engineering: A Practitioner's Approach*
- Sabev, P. & Grigorova, K. (2019). A survey on state of software quality assurance in Bulgaria. In *Proceedings of the 20th International Conference on Computer Systems and Technologies (CompSysTech '19).* Association for Computing Machinery, New York, NY, USA, 124–130.
- Valle-Gómez, K. J., Delgado-Pérez, P. Medina-Bulo, I. & Fernández, J. M. (2019). Software testing: cost reduction in industry 4.0. In *Proceedings of the 14th International Workshop on Automation of Software Test (AST '19)*. IEEE Press, 69–70.
- Meriah, I. & Raba, L. B. A. (2018). A survey of quantitative security risk analysis models for computer systems. In *Proceedings of the 2nd International Conference on Advances in Artificial Intelligence (ICAAI 2018).* Association for Computing Machinery, New York, NY, USA, 36–40.

Discussion (25 points)

Portfolio Project (300 points)

OPTION #1: Your Own UML Diagram

Your Portfolio Project for CSC505 consists of the following:

- Module 4 Milestone (due in Module 4)
- Module 7 Milestone (due in Module 7)
- Lessons Learned Reflection (from all Modules)
- Final Project

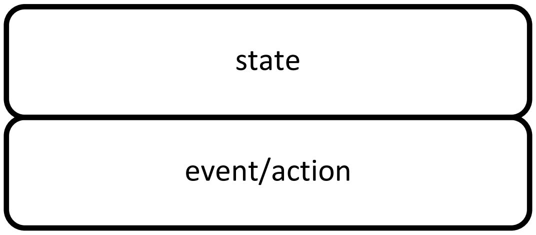
In Week 8, the components left to complete for your Portfolio Project are the **Lessons Learned Reflection** and the **Final Project**. Carefully review the requirements below:

Lessons Learned Reflection
Write a two to three-page summary that outlines the lessons you have learned in this course. Reflect on how these lessons can be applied to real-world problems or to a specific real-world application. How have they affected your life?

Final Project
For your final project, you are going to create a UML diagram of your choice (e.g., Sequence, Class, Activity, etc.) for an automated teller machine (ATM). Your diagram should include the following:

- The customer must pass authentication before withdrawing money.
- Authentication is performed by checking a PIN.
- The PIN can be correct or not.
- Unsuccessful attempts are counted.
- If the counter exceeds a limit, then the customer is rejected.
- If the account balance is zero, then the account is closed.

Annotate your state box in one of two ways. For internal state transitions, use the following state box:

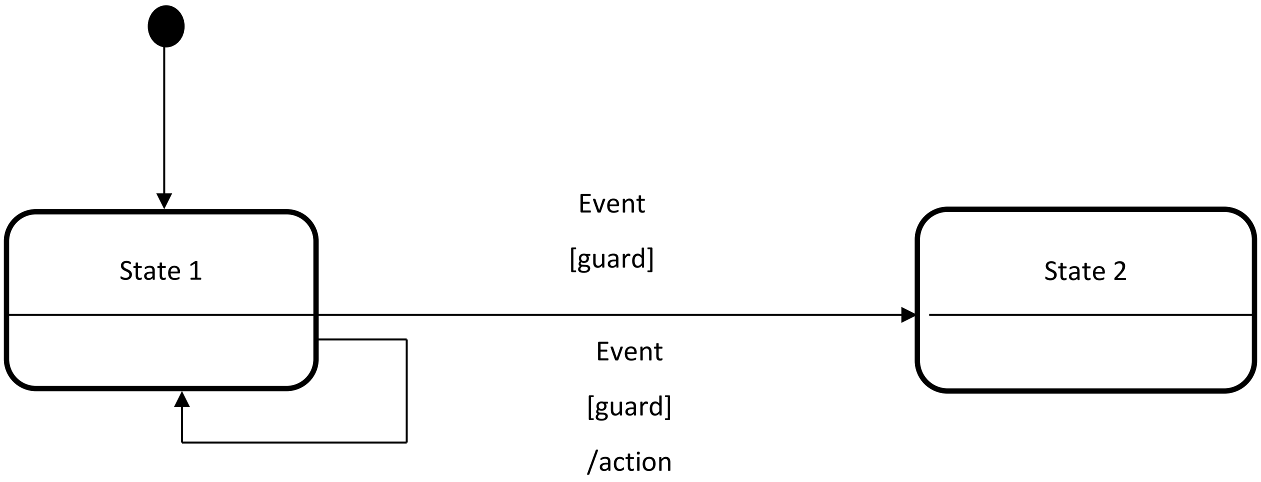


If a state transitions to itself, use the following notation in the state box:

- entry/Action
- exit/Action.

A transition from one state to another is a link arrow connecting the two states and is labeled with **event**, **guard**, or **action**. The “event” is the action that causes a state to transition to another; e.g., “CheckPin.” The “guard” is a possible outcome of an event; e.g., “correct PIN” or “incorrect PIN”. The “action” is the result of the outcome of an event; e.g., if the guard is “incorrect PIN” then the “action” may be something like “increment error counter.” Format your transition labels as follows:

- Event [guard] / action



Use one of the following UML tools (links below) to create your diagrams for the Final Project:

- UMLet
- Gliffy

Your reflection paper should be 2-3 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Submit the final project source file (.py), the screenshots of a successful execution of your program, and all related diagrams, and the Lesson Learned Reflection paper in a zipped folder.

Links:
UMLet
Gliffy

OPTION #2: State Machine Diagram

Your Portfolio Project for CSC505 consists of the following:

- Module 4 Milestone (due in Module 4)
- Module 7 Milestone (due in Module 7)
- Lessons Learned Reflection (from all Modules)
- Final Project

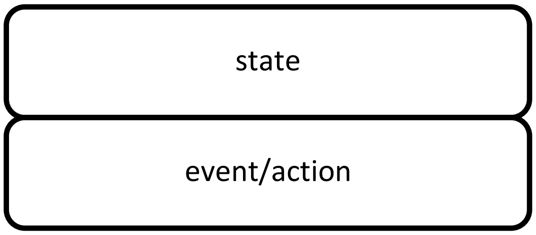
In Week 8, the components left to complete for your Portfolio Project are the **Lessons Learned Reflection** and the **Final Project**. Carefully review the requirements below:

Lessons Learned Reflection
Write a two- to three-page summary that outlines the lessons you have learned in this course. Reflect on how these lessons can be applied to real-world problems or to specific real-world applications. What impact have they had on your life?

Final Project
For your final project, you are going to create a UML State Machine Diagram (SMD) for an automated teller machine (ATM). Your diagram must include the following:

- The customer must pass authentication before withdrawing money.
- Authentication is performed by checking a PIN.
- The PIN can be correct or not.
- Unsuccessful attempts are counted.
- If the counter exceeds a limit, then the customer is rejected.
- If the account balance is zero, then the account is closed.

Annotate your state box in one of two ways. For internal state transitions, use the following state box:

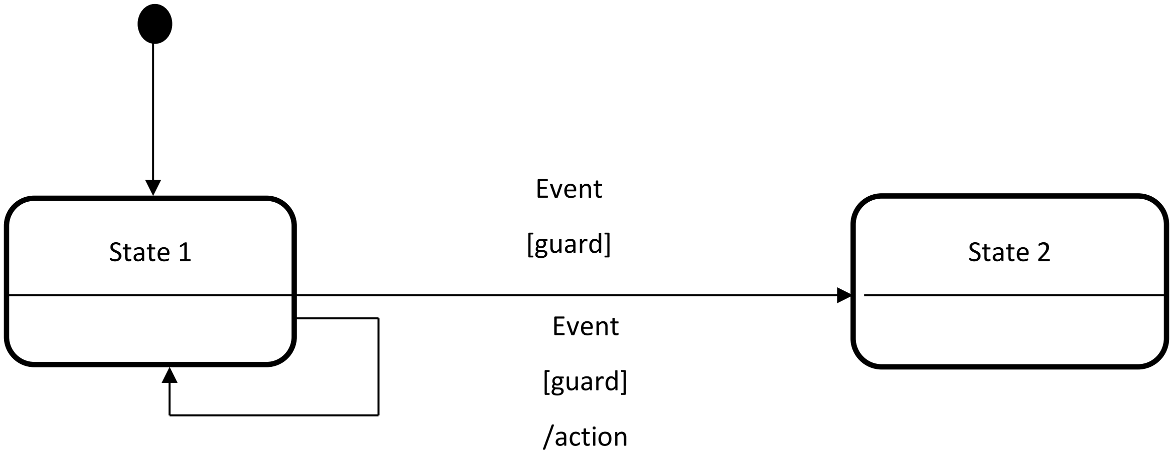


If a state transitions to itself, use the following notation in the state box:

- entry/Action
- exit/Action.

A transition from one state to another is a link arrow connecting the two states and is labeled with **event**, **guard**, or **action**. The "event" is the action that causes a state to transition to another; e.g., "CheckPin." The "guard" is a possible outcome of an event; e.g., "correct PIN" or "incorrect PIN". The "action" is the result of the outcome of an event; e.g., if the guard is "incorrect PIN" then the "action" may be something like "increment error counter." Format your transition labels as follows:

- Event [guard] / action



Use one of the following UML tools (links below) to create your diagrams for the Final Project:

- UMLet
- Gliffy

Your reflection paper should be 2-3 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Submit the final project source file (.py), the screenshots of a successful execution of your program, and all related diagrams, and the Lesson Learned Reflection paper in a zipped folder.

Links:
UMLet
Gliffy

Course Policies

Grading Scale	
A	95.0 – 100
A-	90.0 – 94.9
B+	86.7 – 89.9
B	83.3 – 86.6
B-	80.0 – 83.2
C+	75.0 – 79.9
C	70.0 – 74.9
D	60.0 – 69.9
F	59.9 or below

Course Grading

20%	Discussion Participation
45%	Critical Thinking Assignments
35%	Final Portfolio Project

In-Classroom Policies

For information on late work and incomplete grade policies, please refer to our **In-Classroom Student Policies and Guidelines** or the Academic Catalog for comprehensive documentation of CSU Global institutional policies.

Academic Integrity
Students must assume responsibility for maintaining honesty in all work submitted for credit and in any other work designated by the instructor of the course. Academic dishonesty includes cheating, fabrication, facilitating academic dishonesty, plagiarism, reusing /re-purposing your own work (see *CSU Global Guide to Writing and APA Requirements* for percentage of repurposed work that can be used in an assignment), unauthorized possession of academic materials, and unauthorized collaboration. The CSU Global Library provides information on how students can avoid plagiarism by understanding what it is and how to use the Library and Internet resources.

Citing Sources with APA Style
All students are expected to follow the *CSU Global Guide to Writing and APA Requirements* when citing in APA (based on the APA Style Manual, 6th edition) for all assignments. For details on CSU Global APA style, please review the APA resources within the CSU Global Library under the “APA Guide & Resources” link. A link to this document should also be provided within most assignment descriptions in your course.

Disability Services Statement
CSU Global is committed to providing reasonable accommodations for all persons with disabilities. Any student with a documented disability requesting academic accommodations should contact the Disability Resource Coordinator at 720-279-0650 and/or email ada@CSUGlobal.edu for additional information to coordinate reasonable accommodations for students with documented disabilities.

Netiquette
Respect the diversity of opinions among the instructor and classmates and engage with them in a courteous, respectful, and professional manner. All posts and classroom communication must be conducted in accordance with the student code of conduct. Think before you push the Send button. Did you say just what you meant? How will the person on the other end read the words?

Maintain an environment free of harassment, stalking, threats, abuse, insults or humiliation toward the instructor and classmates. This includes, but is not limited to, demeaning written or oral comments of an ethnic, religious, age, disability, sexist (or sexual orientation), or racist nature; and the unwanted sexual advances or intimidations by email, or on discussion boards and other postings within or connected to the online classroom. If you have concerns about something that has been said, please let your instructor know.