Module 02 Workbook

Missed a day of class or simply want a chance to do some review and refresh what you learned in class? This workbook is for you! We'll be covering the following sections:

- **Learning Objectives**: The goals of the lessons taught
- **Vocabulary**: A list of useful definitions
- **Reading**: A summary of the topics covered in class
- **FAQ**: Some frequently asked questions for learners at this point in the course
- **Learning Resources**: A list of videos and articles that go beyond the in-class content
- **Knowledge Recap**: A short review to quiz yourself

Learning Objectives

After the lessons taught in class, you should be able to:

- Navigate and make changes to your file system using the terminal.
- Break down large problems into sub-problems.
- Use pseudocode to plan your code projects.
- Describe the agile development process.
- Articulate the difference between agile and scrum.

Vocabulary

- **Agile Process**: A development process designed to address some of the waterfall method's pain points. Agile is a cycle that includes these steps:
    1. Meet
    2. Plan
    3. Design
    4. Develop
    5. Test
    6. Evaluate
    7. Repeat
- **Backlog**: In scrum, a list of all the things that still need to be done on a project

- **Development Process**: How you write code, as opposed to the code you write
- **Directory**: Another word for *folder*, just like you would find in your file explorer or finder application
- **Pseudocode**: Pseudocode means breaking down a process into simple steps. This is generally done in plain English, not code.
- **Scrum**: Scrum encompasses specific ideas that align with agile development. Scrum outlines a series of ceremonies and roles which, when utilized, help agile development processes.
- **Sprint**: A one- to four-week period of time in which work is accomplished in scrum
- **Terminal**: A terminal is a computer program that allows us to type instructions. It is used by programmers for many tasks. It does not have a graphical interface, but instead uses only text for its interface.
- **Waterfall Process**: This is a development process in which requirements and instruction come from the top down, with little to no feedback and iteration. The waterfall process is inflexible and often suffers from miscommunication.

Reading

Today's material can be broken into three main chunks: terminal work, pseudocode work, and agile/scrum work. In learning about the terminal, you should have a terminal program already installed from day one. You will begin by learning some beginner commands. These are only the start of what you can do with your terminal, but they are also the main commands that you will be using:

- `ls` stands for *list*. When you type this command and hit ***Enter***, the terminal will list all of the files and directories contained in your current directory.
- `cd` stands for *change directory*. When you type in this command and hit ***Enter***, you will be taken to the root, or top-most, directory of your file structure. More often, you will type `cd directory-name`, where *directory-name* is the name of the directory you want to move to. Type `cd ..` to go up one directory level.
- `mkdir` stands for *make directory*. To use this command, type `mkdir new-directory-name` and hit ***Enter***. The terminal will create a new folder (or directory) for you located inside of your current folder.
- `code .` opens your default code editor in the current directory. For you, it should open up VSCode.

On Mac, you may need to follow these steps to install the `code .` command.

1. Open VSCode.
2. Hit ***Command + Shift + P***.
3. Type **install Code**.
4. Hit ***Enter***.

To learn about pseudocode, learners completed an activity as a class. Take a brief moment (about five minutes) to write down the steps for making a peanut butter and jelly sandwich. While this may seem obvious, each step needs to be in place. Before reading further, try writing the steps down yourself.

How did it go? How many steps did you have? Did you remember to open the jars, or would a robot have had to slice through the lid to get to the peanut butter? Did you open the bread bag and remove your two slices, or was the peanut butter just spread all over the outside of the bag? Little details end up mattering a lot in code. Pseudocode helps us break down each problem into sub-tasks. Decomposition, repetition, and abstraction are all elements of creating pseudocode. For example, you can break down, or decompose, spreading peanut butter on bread into the following steps:

1. Open the jar by unscrewing the lid counter-clockwise, and then place the lid to the side when it comes free.
2. Use the blade of a knife to scoop out approximately 1 Tbsp of peanut butter from the open mouth of the jar.
3. Spread the scooped-out peanut butter onto a piece of bread.

This assumes that earlier steps have been completed correctly. But now, rather than repeating those steps verbatim, you can recognize that a jelly jar is almost the same as a peanut butter jar. This is *abstraction*. It allows you to say, "Repeat the peanut butter steps with the jelly jar," instead of listing out every step again. By breaking down our problem into smaller, more manageable, pieces, you are able to solve complex problems more easily.

Finally, course participants learned about agile development and scrum. There are two main development processes in use today: waterfall and agile. Waterfall has been used for the longest amount of time and thus is largely the default. Essentially, someone wants a project completed, so they explain their requirements to the developers all at once and then wait for the developers to finish. This process is inflexible and can cause problems. Agile development, on the other hand, is a development process that emphasizes flexibility. Rather than receiving requirements all at once, the process is focused on a cycle of working through requirements, then evaluating and updating the next set of requirements to work on. This process allows the project owner and developers to be in constant communication, providing further flexibility and versatility.

You should spend some time going through the asynchronous lesson to learn about scrum and how it works.

FAQ

Question: What is the terminal, and why should I use it?

Answer: The terminal is a text-only interface that allows you to control your computer. It gives you granular control over tasks. Most importantly, there are some tasks which can only be completed from the terminal, so getting comfortable using it now will make those later tasks easier.

Question: Why should I care about pseudocode?

Answer: As you start working on bigger projects and more intense code, writing out the steps for thinking through the problem will help you break down the problem. These steps can be written in your code using comments, and they will also allow you to read your code later and understand it more easily.

Question: Is agile or waterfall development better?

Answer: This is a highly debated topic. The agile development process is recommended because of the flexibility and autonomy that it allows developers, but both processes are widely used today. Your eventual place of work may use either process or a hybrid, so it's best to be aware of both.

Question: How is scrum different from agile?

Answer: Agile development describes a development process and includes ideas like the Agile Manifesto. Scrum is a particular implementation of agile development which prescribes a series of rituals and roles. Although scrum is a very popular way of implementing agile development (leading to the terms sometimes being conflated), it is not the only possible way of doing so.

Question: I still don't feel like I understand a topic we covered in class. What should I do?

Answer: We recommend the following order of operations if you need help with something.

1. Consult your class materials to see what's covered on the topic you're struggling with. We generally only ask you to write code that has been covered in lectures.
2. Search the internet for answers on sites like Stack Overflow, Mozilla Developer Network, and W3Schools.
3. Ask your classmates and friends how they have approached the issue.
4. If the above doesn't help, ask the question during class so that your classmates can also benefit from the answer.
5. Ask a question on your class's Slack channel (or in Zoom chat if during class).
6. Ask your associate instructor to explain the issue in a direct Slack message.
7. Show up to your instructor's office hours.

Learning Resources

Videos

Recommended

- Command Line Interface Tutorial (with GitBash) (From: Classsed)

  - This video (1:50–6:50) goes over the usage of the terminal (Git Bash) commands we taught you, plus the `touch` command. Although the video shows the Git Bash terminal being used, the same commands will work identically in Mac or Linux terminals.
- What Is Pseudocode and How Do You Use It? (From: Codeacademy)
  - This video gives a more thorough explanation of what pseudocode is and why you should use it, largely in the context of writing actual code. Don't worry if the syntax (code) is confusing. Try to follow the ideas presented, as they may be helpful later on.
- What's the Difference Between Agile and Scrum? (From: Beyond20LLC)
  - This video is a quick review of agile development, scrum, and the differences between the two.

Readings

Required

- Manifesto for Agile Software Development (From: Agilemanifesto)
  - This is the manifesto written by the creators of the agile development process. You should take some time to review this manifesto and identify how it addresses the pain points of the waterfall development process.

Recommended

- How to Start Using the Terminal to Be More Productive (From: FreeCodeCamp)
  - This article details many terminal commands that may come in handy for you, including ones that weren't covered in class. Sometimes seeing the same commands explained in a different way can be helpful.
- Pseudocode 101: An Introduction to Writing Good Pseudocode (From: Towards Data Science)
  - This article gives a thorough explanation of how to write pseudocode, as well as some more formal techniques for doing so. It explains several reasons why pseudocode is helpful, as well as places where you may run into it professionally. Be aware that the examples given are going to look different from the pseudocode that you write. As you get more experienced and more comfortable with code and comments, the more formal pseudocode format will also seem more natural.
- Scrum: Learn How to Scrum with the Best of 'Em (From: Atlassian)
  - This article gives an in-depth look at scrum, including its terminology and many of the details that sometimes get overlooked. It also includes several in-line video resources that help to explain concepts, and it gives a very brief summary of kanban, which is another popular agile implementation.

Finally, course participants learned about agile development and scrum. There are two main development processes in use today: waterfall and agile. Waterfall has been used for the longest amount of time and thus is largely the default. Essentially, someone wants a project completed, so they explain their requirements to the developers all at once and then wait for the developers to finish. This process is inflexible and can cause problems. Agile development, on the other hand, is a development process that emphasizes flexibility. Rather than receiving requirements all at once, the process is focused on a cycle of working through requirements, then evaluating and updating the next set of requirements to work on. This process allows the project owner and developers to be in constant communication, providing further flexibility and versatility.

You should spend some time going through the asynchronous lesson to learn about scrum and how it works.

FAQ

Question: What is the terminal, and why should I use it?

Answer: The terminal is a text-only interface that allows you to control your computer. It gives you granular control over tasks. Most importantly, there are some tasks which can only be completed from the terminal, so getting comfortable using it now will make those later tasks easier.

Question: Why should I care about pseudocode?

Answer: As you start working on bigger projects and more intense code, writing out the steps for thinking through the problem will help you break down the problem. These steps can be written in your code using comments, and they will also allow you to read your code later and understand it more easily.

Question: Is agile or waterfall development better?

Answer: This is a highly debated topic. The agile development process is recommended because of the flexibility and autonomy that it allows developers, but both processes are widely used today. Your eventual place of work may use either process or a hybrid, so it's best to be aware of both.

Question: How is scrum different from agile?

Answer: Agile development describes a development process and includes ideas like the Agile Manifesto. Scrum is a particular implementation of agile development which prescribes a series of rituals and roles. Although scrum is a very popular way of implementing agile development (leading to the terms sometimes being conflated), it is not the only possible way of doing so.

Question: I still don't feel like I understand a topic we covered in class. What should I do?

Answer: We recommend the following order of operations if you need help with something.

1. Consult your class materials to see what's covered on the topic you're struggling with. We generally only ask you to write code that has been covered in lectures.
2. Search the internet for answers on sites like Stack Overflow, Mozilla Developer Network, and W3Schools.
3. Ask your classmates and friends how they have approached the issue.
4. If the above doesn't help, ask the question during class so that your classmates can also benefit from the answer.
5. Ask a question on your class's Slack channel (or in Zoom chat if during class).
6. Ask your associate instructor to explain the issue in a direct Slack message.
7. Show up to your instructor's office hours.

Learning Resources

Videos

Recommended

- Command Line Interface Tutorial (with GitBash) (From: Classsed)
  - This video (1:50–6:50) goes over the usage of the terminal (Git Bash) commands we taught you, plus the `touch` command. Although the video shows the Git Bash terminal being used, the same commands will work identically in Mac or Linux terminals.
- What Is Pseudocode and How Do You Use It? (From: Codeacademy)
  - This video gives a more thorough explanation of what pseudocode is and why you should use it, largely in the context of writing actual code. Don't worry if the syntax (code) is confusing. Try to follow the ideas presented, as they may be helpful later on.
- What's the Difference Between Agile and Scrum? (From: Beyond20LLC)
  - This video is a quick review of agile development, scrum, and the differences between the two.

Readings

Required

- Manifesto for Agile Software Development (From: Agilemanifesto)
  - This is the manifesto written by the creators of the agile development process. You should take some time to review this manifesto and identify how it addresses the pain points of the waterfall development process.

Recommended

- How to Start Using the Terminal to Be More Productive (From: FreeCodeCamp)
  - This article details many terminal commands that may come in handy for you, including ones that weren't covered in class. Sometimes seeing the

same commands explained in a different way can be helpful.

- Pseudocode 101: An Introduction to Writing Good Pseudocode (From: Towards Data Science)
    - This article gives a thorough explanation of how to write pseudocode, as well as some more formal techniques for doing so. It explains several reasons why pseudocode is helpful, as well as places where you may run into it professionally. Be aware that the examples given are going to look different from the pseudocode that you write. As you get more experienced and more comfortable with code and comments, the more formal pseudocode format will also seem more natural.
- Scrum: Learn How to Scrum with the Best of 'Em (From: Atlassian)
    - This article gives an in-depth look at scrum, including its terminology and many of the details that sometimes get overlooked. It also includes several in-line video resources that help to explain concepts, and it gives a very brief summary of kanban, which is another popular agile implementation.

Knowledge Recap

Question 1

What does the `cd` command do?

○ list
○ change directory
○ touch
○ cat
○ code .

Question 2

Which of these is NOT one of the reasons pseudocode is useful?

○ It allows us to write out our solutions in plain English.

- It helps us think through solutions.
- It helps identify logic errors.
- It tells the interpreter what to do.

Question 3

The agile development process replaced which common development process?

- Scrum
- Kanban
- Waterfall
- River
- Top-down

Question 4

Describe scrum, including what it is and where it is used.

Show Sample Answer