

[1. General Guidelines 3](#_Toc49782622)

[1.1 General Expectations 3](#_Toc49782623)

[1.2 Timebox 3](#_Toc49782624)

[1.3 Timeline 3](#_Toc49782625)

[1.4 Sending Your Completed Response 3](#_Toc49782626)

[1.5 Language/Environment 4](#_Toc49782627)

[1.6 Online Resources 4](#_Toc49782628)

[1.7 Asking For Help 4](#_Toc49782629)

[2. Requests & Debugging 5](#_Toc49782630)

[2.1 Debugging: Page Doesn’t Load 5](#_Toc49782632)

[3. Development 5](#_Toc49782633)

[3.1 Data Types 5](#_Toc49782635)

[3.2 Code Quality 6](#_Toc49782636)

[3.3 Code Structure 6](#_Toc49782637)

[4. Past Projects 7](#_Toc49782638)

[4.1 Completed Project 7](#_Toc49782639)

[5. Additional Comments 7](#_Toc49782640)

[5.1 Is there anything else you think we should know or you’d like to share? 7](#_Toc49782642)

[6. Coding Samples 8](#_Toc49782643)

[6.1 Custom Sorting 8](#_Toc49782645)

[6.2 Data Validation 9](#_Toc49782646)

[6.3 Simple Web Form 10](#_Toc49782647)

1. General Guidelines
   1. General Expectations

We consider this to be a “full stack” developer test to help us get to know you, your skillsets, and your experience level as well as possible. We hire team members ranging from entry-level to seasoned developers across various areas of the stack, so our goal with this process is simply to determine if we’re a good fit to work together.

You may be comfortable answering all of the questions and if so that’s great! If you feel like some are just not “your thing” that’s fine too. Not every role on our team needs to know *everything* on this test, so just do your best and focus on showing us who *you* are!

* 1. Timebox

This test is intended to be something you can complete in approximately 4 hours depending on skill level. Everyone works at a different pace, so it’s completely acceptable if you want to spend more time, but please don’t spend all day on it.

* 1. Timeline

Generally, we expect to receive your completed response for this test within a week of us sending it to you. If you feel you need more time or something comes up, please let us know.

Once our team receives your completed test, we will generally review and respond within two weeks. At that point, we will either schedule an interview (via phone or in-person) or let you know that we have determined we aren’t the best fit for each other.

* 1. Sending Your Completed Response

For essay questions, please type your written response in the space provided. Each question has at least a half-page allotted for consistency. This simplifies formatting but is not an expectation of how much space each individual question should or shouldn’t take, please take as much or little space as you need.

For the questions that require code as part of the response, please send your completed code as a separate code file and not embedded in the Word document. Hosting on a public GitHub repo is also acceptable. Please be sure to note either the filename or the URL in your written answer to the question.

If you are sending completed code through email and not via a URL for a public repository, please:

1. **Do not attach zip files** as they are blocked by our email filters.
2. **Append “.txt” to the filename** of each code file before attaching to the email. (E.g., “validation.js” becomes “validation.js.txt” so it isn’t filtered by our email rules.)
3. **List the filenames of all attachments** so we can verify nothing was blocked by our email filters.

Name your completed Word document in the format of:

**MetroNet Developer Test - Your Name (YYYY-mm-dd)**

Name any file attachments (for code samples) in the format of:

**MetroNet Developer Test - Your Name (YYYY-mm-dd) - Question Name**

(If a question requires multiple files, please name them accordingly.)

Unless otherwise specified, you may simply reply to the email in which we sent you the test and attach the file containing your completed response.

* 1. Language/Environment

We believe that good developers can learn multiple programming languages, so our goal is to test your general ability to code and reason through problems, not quiz you on a specific syntax or library. Unless otherwise specified, feel free to use whatever programming language you feel most comfortable with to complete this test.

* 1. Online Resources

We believe that searching for answers in Google, Stack Overflow, etc. is part of a developer’s daily workflow and knowing where to find an answer is often as important if not more important than knowing it.

If you use any online resources to help you complete an answer please indicate which ones and a short description of your thought process, including any information you feel relevant such as search terms, what syntax you wanted to verify, etc.

We reserve the right to verify your answers with online tools to check for plagiarism, so be sure not to just copy/paste something from Wikipedia.

* 1. Asking For Help

We believe that helping other team members and being willing to ask questions is an important part of a being a developer. If you get stuck or something doesn’t make sense, please reach out! Just email your hiring contact explaining what’s up and we’ll do our best to accommodate.

1. Requests & Debugging
2. 1. Debugging: Page Doesn’t Load

You’ve developed a **(known working)** web application that, when accessed, will dynamically retrieve a set of data from an SQL database and display it as HTML. When you visit the URL of the application, the page is blank. Describe your troubleshooting steps.

I begin by opening the web dev tools (F12) on a browser and use the ConsoIe tab to see if any errors populated. Assuming the errors are nonspecific, I would check data visual tools provided that checks the health of the program and physical hardware, if available to check for load issues. Assuming none, I would access the server and try to attempt the site locally. If this works, then I begin going through the networking component with network security to find why outside traffic can’t see the service (DNS or firewall issue). If the site was inaccessible locally, I would login to the database to validate that it’s accessible using the same credentials, preferably, to ensure that it’s accessible. From there, I may reboot the server(s). I consider this last part lazy. I’d much rather find the issue to prevent a future occurrence or at least isolate the occurrence to better understand it.

1. Development
2. 1. Data Types

Describe the differences, similarities, and common development pitfalls between: null, an empty string, the number 0, a string containing only the number 0, and Boolean false.

Null is an empty data value, regardless of data type. An empty string denotes a value length of zero for a string data type. The number 0 represents a single character with a data value of “0”, presumably, for a number type. A string containing the number 0 is the single character of “0” with a string data type. A Boolean false typically contains the value of “FALSE” or “0” in the database with a Boolean data type. A null value is quite common for unrequired fields where the user wasn’t required to respond. However, it may be valuable to see an empty string to validate a response was received despite being an empty string. That way it’ll be easier to troubleshoot errors between something not received properly, with a null vs a string response or an empty string.

As far as the difference between int/number “0”, Boolean “0” or “FALSE”, and string “0”, it all comes down to data type. If the field only has 2 values, then Boolean is the way to go. If I want to perform analysis on the numbers, then I would need uniformity across the data values which can be achieved by using a number data type and front-end validation to help prevent the user from putting a string. Lastly, a string should be used when dealing with a mix bag field that isn’t always a number. It’s best to fit the data type with the expected response to perform analyses later to help support business understanding, whether in data visual reporting or creating AI generated models. It’s always best to have clean data.

* 1. Code Quality

How would you describe "good code"? What role (if any) do comments, refactoring, and code reviews play?

Everyone has different styles and it’s important to help conform to a consistent style across the company. To help usher this, having mutually agreed to programming standards is a good start. Taking these standards and adding them into SonarQube, or lint equivalent, is even better.

Good code must always achieve it’s intended goal without causing spillover issues. Any new code adds to the complexity of a system, and it’s important to have comments that help describe the changes accurately, whether in line or in Git. I personally prefer brevity with comments in line and greater extrapolation in Git to avoid scrolling for long periods in the IDE with little actual code in between. Over time the quick fixes can become redundant in purpose instead of reused or given much consideration for application speed. Often times programmers don’t get to just clean up code without new features being added to alleviate tech debt. It’s important when adding a new item to not just add the feature without regards to the rest of the component. If any items can be consolidated and reused within that section of the code without other dependencies, then refactoring for simplicity and speed, marginal, may be warranted.

I love code reviews despite potentially getting heated at times. It’s an opportunity to get feedback to become a better programmer and another point of view to check for weakness prior to potential embarrassment after it’s been promoted to production. It helps teams form together to develop a more unified programming standard along with cross training. From an manager vantage point, it ensures teams are communicating within and reduces the number of costly mistakes due that may cumulatively cause a failed go live resulting in a roll back. Issues may occur but it’s generally better to use switches of features and then roll forward, if the switch didn’t work as expected, at the next release.

* 1. Code Structure

What criteria do you have for deciding when a single method/function does "too much" and needs to be broken down into multiple methods/functions?

This builds well on to the refactoring question, earlier. It’s certainly at the team/programmer’s discretion. However, the more complex a method, the harder it will be for any other programmer to understand and add to. A function, ideally, should be targeting a specific goal or allow for abstraction with generalized layers to reduce repeated code. Although it could be measured by lines of code, I typically draw the line at when it is no longer targeting the exact goal set out by the function. If creating a new object, and the object consists of a monstrous function, then I may break the function apart into unique features, given that this area of the code doesn’t have significant dependencies elsewhere increasing my chances of spillover errors and the time to do so. Casually refactoring a complex billing system may result in numerous unanticipated consequences with substantial time needed for functional and integration testing, which may not be available.

1. Past Projects
   1. Completed Project

Describe the project you have worked on that you are most proud of. What was your part in the project that worked out particularly well?

I enjoyed working on the Auto Vehicle Locator (AVL) tool that was largely developed in house, while at Vectren. I helped develop enhancements in to the tool allowing field dispatchers better understanding of who to call regarding a natural gas leak incident. Technicians share vehicles, have unique union rules, and take time off (vacation, doctor appointment, etc.). I helped work with the team responsible for technician’s work availability schedule to understand their data, access the data, and then use it to help determine the list of technicians to call and the priority based upon proximity of tech’s vehicle location using Google API for travel time (tech to vehicle relationship created in form completion application and sent to AVL) while also applying schedule of availability and union rules. It helped reduce the amount of time to connect a dispatcher with a technician to ensure regulatory response within 30 minutes of tech on site and customer call. Prior to when I left, the average was 18-20 minutes which was much lower than many of our peers. I enjoyed the collaboration with another programming team, and several business units to make this work, while also helping the community around me.

5. Additional Comments

1. 1. Is there anything else you think we should know or you’d like to share?

Before leaving Vectren/CenterPoint for OMF, I was the interim QA manager. I was responsible for that department’s budget, 2 FT contractors, QA related audits, and assisting in QA merger requests while also maintaining all previous applications. I left Vectren for OMF to join the newly formed Emerging Technologies team in pursuits of greater technical development growth as CenterPoint planned to scrap nearly all inhouse development to focus on more canned solutions. After Covid hit, the OMF decided to focus on core technologies and dismantled this team. I still enjoy coding and trying new ways to solve problems in efforts to find a better solution (e.g., increase performance, reduce technical debt etc.).

6. Coding Samples

1. 1. Custom Sorting

**Task:** Write a function to sort a hand of cards.

**Input Parameters:** a list/vector/array of Card objects

**Return Parameter:** a list/vector/array of Card objects that are sorted

**Assumptions you don’t have to code:**

* Each card object has an attribute called **suit** that returns the suit of the card as a string: “Hearts”, “Spades”, “Clubs”, “Diamonds”
* Each card object has an attribute called **value** that returns a character representing their value: 2, 3, 4, 5, 6, 7, 8, 9, J, Q, K, A

**Requirements:**

* The cards should be sorted in **ascending** order unless you implement the bonus flag below. In that case, the default should be ascending unless overridden by the flag.
* **Bonus:** Add a flag to say whether to sort in ascending or descending order.

**Domain Knowledge:**

* Playing cards are ordered by **value** then by **suit**.
* For values, assume: 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9 < J < Q < K < A
* For suits, assume: Hearts < Diamonds < Clubs < Spades

**Example of Sorted Cards:**

1. 3 of Clubs
2. 7 of Hearts
3. Ace of Hearts
4. Ace of Spades

**Context/Hints:**

* Try to treat this like you would any other real-world sorting problem you may encounter in normal business logic.
* Feel free to write additional helper functions or other functions associated with the Card object to help accomplish the task.
  1. Data Validation

**Task:** Write a simple script to validate a set of contact records and report on any errors.

**Given:**

* A list of 20 contact objects (full names, city, phone number, and email address)

**Step 1: List all contact records with the following output:**

* Full name
* Whether the phone and email fields are "valid":
  + Output "Valid" if both email and phone are valid.
  + Output "Email is invalid." if email is invalid and phone is valid.
  + Output "Phone is invalid." if phone is invalid and email is valid.
  + Output "Email and Phone are invalid." if both phone and email are invalid.

**Step 2: List each city and report the following output:**

* Name of city
* Number of validation errors

**Requirements:**

* Contacts should be sorted alphabetically in **ascending** order.
* Cities should be sorted by number of validation errors in **descending** order.

**Validation criteria:**

* Email field: has exactly one @ symbol with data on each side
* Phone field: is numeric with only digits, dashes, and spaces allowed

**Data Set:** Use the records in Contacts.json, which are based off of U.S. census data via Wikipedia’s list of [given names](https://en.wikipedia.org/wiki/List_of_most_popular_given_names) and [surnames](https://en.wikipedia.org/wiki/List_of_most_common_surnames_in_North_America).

* 1. Simple Web Form

**Requirements:**

* Page title should be “Team Introduction”.
* Page should have a simple HTML form that requests two fields of input: your name and a fun fact about yourself.
* The form should have a button to introduce yourself.
* When the submit button is clicked, both input fields should be validated and an alert message shown if there is an error.
* If there are no errors when the form is submitted:
  1. The validated data should be logged to the browser’s developer console.
  2. The form should be hidden and replaced with a new box displaying the input (name and fun fact).
* The “introduction box” should have a link or button to reset the form and allow “introducing” a different team member.
* **Bonus:** Use SCSS or describe how you’d refactor your CSS if SCSS was available in the build environment.