**TASK 1 – PROGRAM PLANNING**

In this task you will prepare for your program writing by doing the following:

A.  Create a flowchart for a program to perform the required task.

B.  Write pseudocode for a program to perform the required task.

C.  Provide an explanation of the relationship between the flowchart and pseudocode that does the following:

1.  Describe the logic behind the flowchart and pseudocode.

2.  Explain the alignment between flowchart and pseudocode.

**TASK 2 – CODING**

In this task you will create the following program in Gitlab:

A.  Create a program in Gitlab using Python or R to perform the data analysis described in Task 1.

**TASK 3 – PRESENTATION**

In this task you will create a report for your stakeholders by doing the following:

A.  Explain how the code works for the program you submitted in Task 2.

B.  Provide **4** customized data visualizations.

C.  Explain how customized visualizations in part B were created.

**COURSE INSTRUCTOR TIPS FOR SUCCESS**

These tips provide my suggestions as well as answers to the most common student questions. Tips are organized alphabetically within each task requirements section (e.g. A, B, C, etc.) and within the *General*section, which comes before the task requirements sections.

General Suggestions

**Basic Tips**

1. Complete the self-assessment found via the *Go to Course Material* button on the course page, then select *Section 1* > *Course Overview* > *Before you start the course*.

2. Performance assessment 1 is creating a flowchart. No programming is required in performance assessment 1.

3. Performance assessments 2 and 3 require writing a program in either Python or R. Use the same programming language for both performance assessments. You are free to choose either language and should consider the industry you work in or intend to work in as well as the overall job market.

4. Practice coding by duplicating the examples you study. Adding comments in your own code that explain what the code is doing is a great way to learn.  
  
  
**GitLab**

Ensure you correctly create your GitLab course specific branch by following [the instructions in a 3-minute video](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8a75f1be-fe95-4031-8f76-b22700fe8716&openInDeviceBrowser=true).

**CAUTION:** Do not use the "New project" button in the upper right corner of your repository in GitLab. This button causes downstream problems. WGU and GitLab are working on removing this button. You should use the + button in the middle of your GitLab repository page to create new directories, new branches, and upload files.

You should use your WGU email address to create your account in GitLab. This allows instructors to find your repository when you ask for help.

GitLab is a repository for managing version control and running pipelines. You can read more about [CI/CD Pipelines](https://docs.gitlab.com/ee/ci/pipelines/). However, you will not use CI/CD Pipelines in this class.

We are using GitLab in this class as the preferred method for you to submit your code to the Evaluation Team. You do this by submitting the URL to your GitLab repository branch.

First, create your class-specific repository branch by following [the instructions in a 3-minute video](https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=8a75f1be-fe95-4031-8f76-b22700fe8716&openInDeviceBrowser=true).

Then, code your solution on your local machine in the IDE (Integrated Development Environment, e.g. Jupyter Lab, R Studio, etc.) of your choice and run, debug, and perfect your code on your local machine.

Finally, upload your code to your class specific repository branch using the + button (see time 1:35 in the 3-minute video linked above). Then submit the URL of your GitLab branch containing your code file to the Evaluation Team via the task submission process by [following these instructions](https://cm.wgu.edu/t5/Information-Resources/Performance-Assessment-Submission/ta-p/3714).

D600 Statistical Data Mining has tasks requiring you to "commit with a message and push [to GitLab] when you complete each requirement" of the task. Thus, learning to do this now is a good strategy. Follow the version control suggestions in D598 > Go to Course Material > Section 3 > Lesson 2: GitLab > Section 2: Working with Version Control [a 5-minute video] to commit to GitLab with a message when you get to D600.

Task 1:

A & B - Ensure you include ***decisions***

Ensure you include logical decision-making in your flowchart and pseudocode. A logical-decision must be made any time the process requires a *yes*or *no*answer to a question. For example, do records match a specific criterion? If *yes*, do one thing. If *no*, do another thing. The flowchart symbol for a decision is the diamond, and is illustrated in this article: <https://www.geeksforgeeks.org/an-introduction-to-flowcharts/>

Task 2:

A - Writing your code

The task requirements state, "Create a program in GitLab...". However, you will not create your program in GitLab. You will upload you program (code file) to GitLab. See the GitLab tip above for more details.

Do not include a file path in your code. Instead, read the Excel data file from the same directory as the code file so that no file paths are included in your code.

You are expected to follow the coding best practice of preventing divide-by-zero errors when calculating the debt-to-income ratio. For any business with zero income, I suggest assigning a debt-to-income of 0% if the debt is 0 and a debt-to-income of 100% if the debt is greater-than 0. There are no businesses with zero income in the provided data. However, coding to prevent divide-by-zero errors is a coding best practice. [Geeks-for-geeks](https://www.geeksforgeeks.org/how-to-return-0-with-divide-by-zero-in-python/?openInDeviceBrowser=true) has a good explanation with several approaches to preventing divide-by-zero errors.

The WGU MSDS QA virtual lab is available at <https://platform.qa.com/lab/wgu-assessment-lab/>. (This link is correct, and it returns a 404 error until you log in the the QA WGU subdomain. Select the "Go back to login" link on the 404 page, then select the "Login with your company account" link, then enter "WGU" as the company subdomain and select "Continue".) The company called QA was formerly Cloud Academy. This virtual lab is available for coding your solutions to the tasks. QA has an internal help feature, chatbot, and customer support. See the [WGU Knowledge Center support article for QA](https://cm.wgu.edu/t5/Frequently-Asked-Questions/Cloud-Academy-Labs/ta-p/62862) for details on these and more features of QA. However, you are not required to use this virtual lab. You may use any IDE on any platform you choose. You are required to code your solution in either Python or R. Many students use [Anaconda](https://www.anaconda.com/) with [Jupyter Notebook](https://jupyter.org/?openInDeviceBrowser=true) for Python and [RStudio](https://posit.co/products/open-source/rstudio/) for R. You are not required to use any specific IDE.

W3Schools has great resources for learning about and practicing both [Python](https://www.w3schools.com/python/default.asp) and [R](https://www.w3schools.com/r/default.asp).

Task 3:

B - Visualizations (graphs, charts, etc.)

The [Python Graph Gallery](https://www.python-graph-gallery.com/) and [R Graph Gallery](https://www.r-graph-gallery.com/) are each the single best resource for visualizations for each tool. Both sites are beautifully designed with an easy visual search home page and navigation bar and provide detailed examples and suggestions for coding and improving your visualizations.