MATH 282 Assignment #2

Computational methods for integration

**Due by 10pm on Monday, October 20th, 2025**

**Total marks: 28**

# Submission Instructions

If you have a paper copy (handwritten) of any of your answers, submit the paper copy to Michael, or scan or photograph the paper copy to be submitted.

* Your names and SaskPolytech usernames must be on the pages
* Answers should be clearly labelled with the question number
* Answers must be legible

For submitting files:

* Create a folder named ***username1username2*math282a2**, where ***username1username2*** are the SaskPolytech usernames of your group members in alphabetical order.
* Create subfolders for each question named **q1**, **q2**, etc.
* Place the files for each question in the subfolder for that question.
* For programming questions:
  + For Java programs, include only the source code (**.java** files).
  + Make sure that all files have proper comments and follow the CST programming style guide
  + Ensure that all source code files are submitted (so the program can be compiled and run from the files submitted)
* When done, create a compressed folder named ***username1username2*math282a2.zip**
* Submit the compressed (zipped) file to the Assignment #2 Dropbox Folder in the MATH282 online course material for MATH 282 (under Assessments > Dropbox).

1. *[7 marks]* In the **Assignments\a2q1** folder in the MATH282 OneDrive folder, you will find Java files with starting code for this question for the function class. Copy these files into IntelliJ (or your preferred development environment) and run **A2Q1TestFunction** to see the current state of the program.

Delete the code currently in the **main** method in **A2Q1TestFunction** (you can also delete the files **FunctionExample.java** and **FunctionSpeed.java**, since they were only needed for testing). In the **main** method of **A2Q1TestFunction** (and adding/modifying other classes as needed), use the left rectangle rule to calculate and display the following integrals to a relative precision of 0.000001:

*Notes:* log2(*x*) means “the base-2 logarithm of *x*” – in other words, the power of 2 that produces the result *x*, or the value *y* such that 2*y* = *x*. For instance, log2(1024) = **10**, since 2**10** = 1024.

Most programming languages have a log method, but the method may not allow you to specify the base; it may always do natural logarithms (to the base *e*), or base-10 logarithms. To create base-2 logarithms from any other base, use the following formula:

log2(*x*) = log(*x*) / log(2)

You can check that your log2(*x*) function works by comparing it to Excel’s **LOG** function. The Excel formula **=LOG(*value*, *base*)** will calculate the logarithm of a given value to a desired base. For instance, **=LOG(1024, 2)** results in **10**.

1. *[7 marks]* In the **Assignments\a2q2** folder in the MATH282 OneDrive folder, you will find the Excel workbook **A2Q2Calculations.xlsx**. On the **Documentation** worksheet, fill in cells C2, C3, and C20 with your name(s), username(s), and the requested result from question #1(c) *[if you are already done question #1(c) – if not, you can start question #2 and go back and fill in the result later, or use the result from another source like Wolfram|Alpha]*.

From the **Assignments\a2q2** folder in the MATH282 OneDrive folder, print the Word document **A2Q2Graphs.docx** (or just edit the document directly). Answer the questions on the printout (or in the edited document), filling in the appropriate worksheets in the Excel workbook where requested.

1. *[14 marks]* In the **Assignments\a2q3** folder in the MATH282 OneDrive folder, you will find Java files with starting code for the function class and an Integral class. Copy these files into IntelliJ (or your preferred development environment) and run **A2Q3TestIntegral** to see the current state of the program.

In **A2Q3Integral.java** and **A2Q3TestIntegral.java**, add appropriate Javadoc comments to the existing code.

In **A2Q3Integral.java**, the code for the rectangle rule is not efficient. Rewrite the code so that the calculations are done efficiently (using the improvements described and documented in class).

In **A2Q3Integral.java**, the code for the trapezoid rule is not done. Implement the trapezoid rule. For full marks, make the trapezoid rule efficient (using the improvements described in class).

In **A2Q3Integral.java**, the code for Simpson’s rule is not done. Implement Simpson’s rule. Implement any efficiencies that you can find.

In **A2Q3TestIntegral.java**, add code in the **main** method (and adding/modifying other classes as needed) to calculate the following integrals to a relative precision of 0.000001 using each of the rectangle rule, trapezoid rule, and Simpson’s rule. *Note that you can still receive marks for this part of the question if only the rectangle rule is working!*

* 1. The integral (or integrals, for groups of students) assigned to you in **A2Q3Integrals.docx** in the **Assignments\a2q3** folder in the MATH282 OneDrive folder. Each group need to complete 3 based on the group# column.

From the **Assignments\a2q3** folder in the MATH282 OneDrive folder, open **A2Q3Output.docx**. Paste your output from your completed version of **A2Q3TestIntegral** in the requested location. Answer the question at the bottom regarding which integration rule is the most efficient.