

Building a Tax Calculation App

Read the requirements thoroughly, answer the questions, and submit your answers in a compressed folder as instructed below.

Requirements:

Design and code a tax calculation app for Ontario based on the tax calculation rules found in “Combined Federal & Ontario Tax Brackets and Tax Rates Including Surtaxes” section of <https://www.taxtips.ca/taxrates/on.htm>.

The app will also account for RRSP contributions (i.e deduct RRSP from income when calculating tax, and display RRSP contribution limit for next year). RRSP limits are found on <https://www.canada.ca/en/revenue-agency/services/tax/registered-plans-administrators/pspa/mp-rrsp-dpsp-tfsa-limits-ympe.html>.

Your app should include a tax calculation class that takes care of everything related to taxation and one activity that shows a proper UI to the user and call the tax calculation methods properly. UI choices are up to your creative imagination. The app should allow its users to do some what-if analysis by providing a [Slider](#) to choose a multiplier (range: \$0 to \$27,230) for the RRSP deduction limit and calculate the tax as the user moves the slider. Here is a tutorial about Sliders: <https://medium.com/analytics-vidhya/sliders-material-component-for-android-5be61bbe6726>.

App data should be stored in Shared Preferences and retrieved whenever the app starts. A refresh button will be added to reload the data stored in shared preferences.

Submission details:

- From Android Studio -> File menu, export your project as zip file, name it StudentID_StudentName.zip
- Copy all of your codes (java and XML files you created/changed) to a new word document, name it as above, and add 3+ app screenshots (showing different screens) to the document.
- Submit both your app zip folder and document by the due date

Rubric

	Unsatisfactory (0-40%)	Satisfactory (40-60%)	Good (80%)	Excellent (100%)
Requirements met 50%	<ul style="list-style-type: none"> Completed less than 50% of the requirements. Does not execute due to errors. 	<ul style="list-style-type: none"> Completed 50-60% of the requirements. Executes without errors. 	<ul style="list-style-type: none"> Completed 70-90% of the requirements. Executes without errors. 	<ul style="list-style-type: none"> Executes properly Excellent User Interface
Solution efficiency 20%	<ul style="list-style-type: none"> A difficult to comprehend UI, or Inefficient code 	<ul style="list-style-type: none"> Solution is good and easy to follow but efficiency can be enhanced, or Disorganized code 	<ul style="list-style-type: none"> Solution is efficient, Code is organized and easy to follow, good UI 	<ul style="list-style-type: none"> excellent UI code
Design and Coding Standards 20%	<ul style="list-style-type: none"> Poor use of space, conventions, colours and standards Disorganized UI or code 	<ul style="list-style-type: none"> Acceptable use of space and colours Most standards and conventions were well followed UI is mostly organized. 	<ul style="list-style-type: none"> Good use of space and colours Standards and conventions were well followed Organized work. 	<ul style="list-style-type: none"> Excellent use of space and colours Standards followed Creatively designed and organized work
Delivery 10%	<ul style="list-style-type: none"> Not delivered in correct format (not submitted online, incorrect format...) No name, date, or title included 	<ul style="list-style-type: none"> Delivered on time, and in correct format (Word/ PDF/Zip etc) Includes name, date, title. 	<ul style="list-style-type: none"> Delivered on time, in correct format Includes name, date, and title 	<ul style="list-style-type: none"> Delivered on time, proper format Includes name, date, and title