Software Development Project

In this unit, you will work on a major programming project in a team of four. It will count 15% of your report card mark, but only half of that will be from programming. The other half will be determined by

* your design (which will be on paper)
* your design of test cases that ensure your program works on a variety of inputs
* your ability to work on a team.

## Learning objectives

Doing this project will help you

* think abstractly about a problem
* design solutions from the top-down, using stepwise refinement (more on that later)
* solidify your math skills
* improve your team work skills

## Topic choices

Your team will choose one of the following topics to do your project on:

**Choice A: A factoring toolbox**

This program lets the user enter a quadratic trinomial of the form *ax*2 + *bx* + *c*, where a, b and c are integers, and outputs either:

* the two factors of the trinomial if it can be factored
* or the statement, “This trinomial cannot be factored” if it can’t

The program should be able to handle:

* Trinomials with a = 1 or a > 1.
* Differences of squares
* Common factoring

**Choice B: A graphing calculator and analyzer for linear and quadratic functions**

This program lets the user enter a quadratic function of the form y = *ax*2 + *bx* + *c* (where a, *b* and *c* might be 0) and outputs the following:

* A graph of the function over a range of x-values that the user specifies. The graph should include x- and y-axes, axis labels, tick-marks, and the equation.
* The roots of the function (i.e. x-values where the graph crosses the y-axis), if any.
* The slope and y-intercept of the function if it is linear
* The vertex and the direction-of-opening for the function if it is quadratic

## What your team will turn in

By the end of the project, your team will turn in a .zip file on Edmodo containing 3 things:

1. **A design document** that describes
   1. The purpose and functionality of the program
   2. Instructions for using the program
   3. A list of all the functions and procedures your program will use
   4. The arguments they take, and their return-values or the output they produce.
   5. Who on your team will be responsible for coding each function or procedure.
   6. A set of at least 10 test cases, with the expected outputs.
2. **A pair of Python programs**:
   1. one that contains your package of functions
   2. one that contains your 10 test cases and runs the program on those cases.

You may begin coding only after I have seen and approved your design document.