

IDBS Coding Assessment Exercise- Python

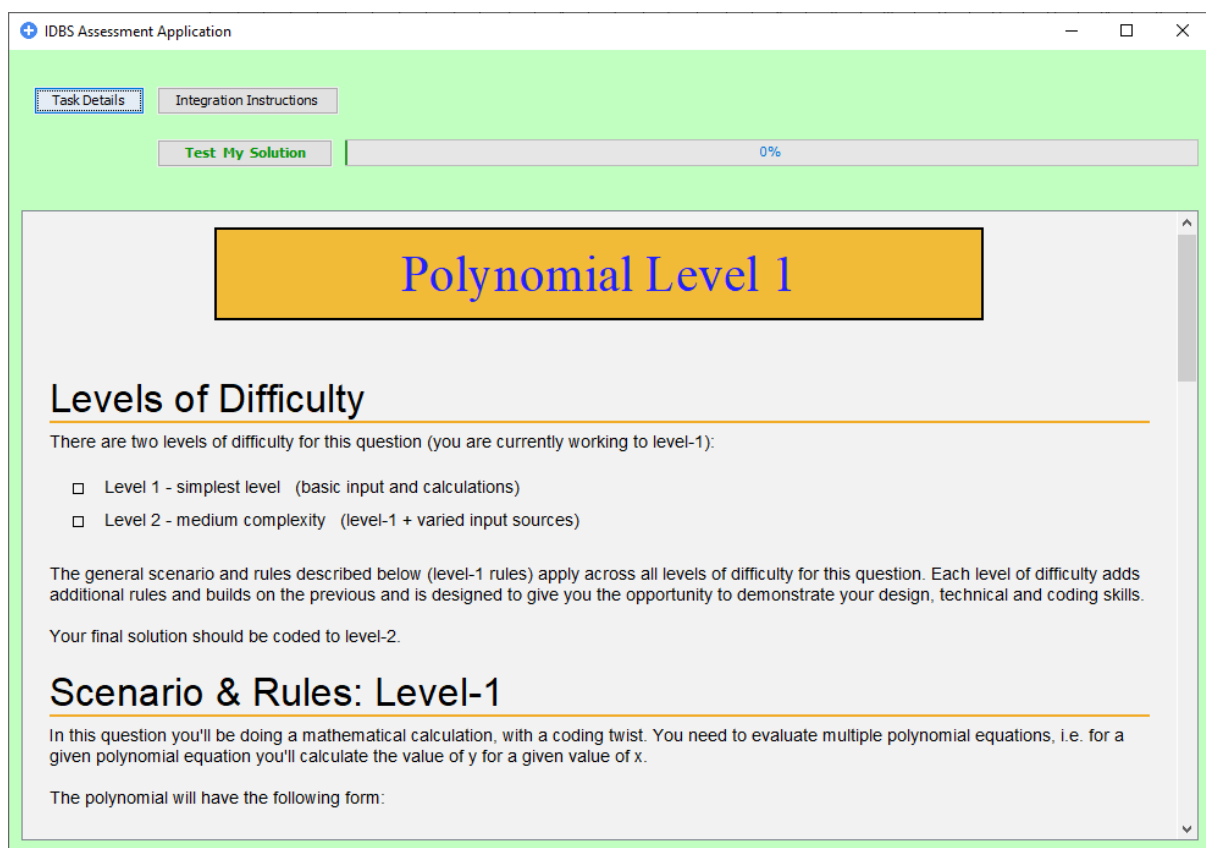
This document describes how to complete the Assessment Exercise using Python (or any other alternative language).

IMPORTANT NOTICE: *The assessment exercise was originally developed as a java based exercise using a java application as the platform. The candidate integrates their java solution into the java app and the java app generates and feeds random data to the candidates coded solution.*

It is feasible to complete the assessment exercise in a language other than java by using the data files provided as an alternative source of random data. The pages provided below are the various instructions pages taken from the java application describing the test scenario and rules for exercise. Other than java specific items all instructions apply within your Python (or alternate language) solution.

To put the various sections below into context an image of the running java application is provided below. Each section below represents the instructions displayed when the candidates presses the **Task Details** and **Integration Instructions** buttons.

All data for the python (alternate language) exercise are available within the zipped text files provided on the repo. The data has been generated from the running java app.



(shown when: **Task Details** button pressed with level-1 difficulty selected)

Polynomial Level 1

Levels of Difficulty

(when coding your solution to level-1 use the Level 1 data files)

There are two levels of difficulty for this question (you are currently working to level-1):

- Level 1 - simplest level (basic input and calculations)
- Level 2 - medium complexity (level-1 + varied input sources)

The general scenario and rules described below (level-1 rules) apply across all levels of difficulty for this question. Each level of difficulty adds additional rules and builds on the previous and is designed to give you the opportunity to demonstrate your design, technical and coding skills.

Your final solution should be coded to level-2.

Scenario & Rules: Level-1

In this question you'll be doing a mathematical calculation, with a coding twist. You need to evaluate multiple polynomial equations, i.e. for a given polynomial equation you'll calculate the value of y for a given value of x.

The polynomial will have the following form:

$$y = 3x^4 + 2x^3 - 6x^2 + 8x - 2$$

- the number of terms in the polynomial will not exceed 10 terms
- the value of the multiplier will not exceed 10
- the value of X will not exceed 50
- all numbers used (the power, multiplier, constant and value of X) will always be positive integers

Now here is the coding twist:

The code that you write to evaluate the equation must ONLY utilise the mathematical operators ADD and SUBTRACT.

You **are not allowed** to use mathematical operators such as multiplication, division, modulus, "raised to the power of", logarithms, base conversions, logical-shift operators or any mathematical functions/operators other than ADD " + " or SUBTRACT " - ". If the code you submit uses any method other than simple ADD or SUBTRACT during the calculation process the solution will be rejected.

Use of the java increment/decrement operator ++ or -- (for example: i++ as a loop counter) is allowed within the code of your solution, but the value should not be used as part of an addition or subtraction within the polynomial calculation itself.

Level-1 Specifics:

The polynomial will be provided as input data in the form of Json as follows, (this json is for the example above):

```
{
  "xValue": 5,
  "terms":
    [
      {
        "power": 1,
        "multiplier": 8,
        "action": "add"
      },
      {
        "power": 0,
        "multiplier": 2,
        "action": "subtract"
      },
      {
        "power": 4,
        "multiplier": 3,
        "action": "add"
      },
      {
        "power": 3,
        "multiplier": 2,
        "action": "add"
      },
      {
        "power": 2,
        "multiplier": 6,
        "action": "subtract"
      }
    ]
}
```

Testing Your Solution

Use the level-1 input files for this scenario. Each input file contains 100 polynomials encoded as a json string with the expected numeric y-value of the polynomial on the following line.

Your solution should read the data files, calculate the result of the polynomial (according to the rules of the exercise) and ensure your value equates to the expected value provided in the file.

Once you have your solution working for all input lines and files AND you are satisfied with the quality of code you have written - **you are ready to submit your solution** - please see *Integration Instructions* for instructions on submitting your solution.

What We're Looking For

In this assessment exercise we are assessing your design and coding skills so your code should be of the highest quality – well designed using OOD/OOP practices (or functional programming techniques), readable, reliable, robust and maintainable. As well as using any techniques that you feel appropriate for high quality and maintainable code you should aim to provide a reasonable level of test coverage for your solution.

When writing unit tests you MAY use mathematical operators other than ADD and SUBTRACT in your test code. .

Third Party Libraries

Please do not use any third party libraries beyond the standard runtime java libraries other than those supplied by us in your solution. For security reasons we will be unable to accept any solution that uses third-party libraries other than those we have supplied (Json api and our Assessment Application implementation libraries - it is **not necessary** to submit these libraries with your solution). If any additional libraries are submitted the solution will be rejected.

(shown when: **Task Details** button pressed with level-2 difficulty selected)

Polynomial Level 2

Levels of Difficulty

(when coding your solution to level-2 use the Level 2 data files)

There are two levels of difficulty for this question (you are currently working to level-2):

- Level 1 - simplest level (basic input and calculations)
- Level 2 - medium complexity (level-1 + varied input sources)

The scenario and rules described below are for level-2 difficulty. The general scenario and rules for level-1 apply for this level but with the additional rules noted - each level builds on the previous and is designed to give you the opportunity to demonstrate your design, technical and coding skills.

Your final solution should be coded to level-2.

Scenario & Rules: Level-2

In this question you solving the same type of polynomial equation as in level-1 with the same limitations on the use of mathematical operators.

As before the polynomial will have the following form:

$$y = 3x^4 + 2x^3 - 6x^2 + 8x - 2$$

- the number of terms in the polynomial will not exceed 10 terms
- the value of the multiplier will not exceed 10
- the value of X will not exceed 50
- all numbers used (the power, multiplier, constant and value of X) will always be positive integers

The coding twist:

The code that you write to evaluate the equation must ONLY utilise the mathematical operators ADD and SUBTRACT.

You **are not allowed** to use mathematical operators such as multiplication, division, modulus, "raised to the power of", logarithms, base conversions, logical-shift operators or any mathematical functions/operators other than ADD " + " or SUBTRACT " - ". If the code you submit uses any method other than simple ADD or SUBTRACT during the calculation process the solution will be rejected.

Use of the java increment/decrement operator ++ or -- (for example: i++ as a loop counter) is allowed within the code of your solution, but the value should not be used as part of an addition or subtraction within the polynomial calculation itself.

Level-2 Specifics:

The format of the input data defining the polynomial may vary in this level.

There are now two forms of input data defining the polynomial and a full test run will use BOTH forms randomly selected at approximately 50/50. **The business analysts driving the requirements for the application have also indicated that other input formats may be used in the future** - however these have not yet been specified and as yet are not included in the test data generated for this level.

The new formats that you are required to support are as follows:

Input Format 1 - json:

This is the same json format used in Level-1 thus:

```
{
  "xValue": 5,
  "terms":
  [
    {
      "power": 1,
      "multiplier": 8,
      "action": "add"
    },
    {
      "power": 0,
      "multiplier": 2,
      "action": "subtract"
    },
    {
      "power": 4,
      "multiplier": 3,
      "action": "add"
    },
    {
      "power": 3,
      "multiplier": 2,
      "action": "add"
    },
    {
      "power": 2,
      "multiplier": 6,
```

```

        "action": "subtract"
    }
]
}

```

Input Format 2 - numeric:

This is a new format and uses a highly compact numeric & mathematical-notation. This format has a prefix string indicating the new input type thus:

```
numeric:x=5;y=+8.x^1-2.x^0+3.x^4-6.x^2+2x^3
```

This format uses:

a dot . to mean multiply or times
the hat ^ to mean "to the power of"

hence the input data above can be read as:

x equals 5

y equals 8 times X to the power of 1
minus 2 times X to the power of zero
plus 3 times X to the power of four
minus 6 times X to the power of 2
plus two times X to the power of three

note: the plus or minus sign will ALWAYS be
present on the multiplier of the first
term even if that multiplier is zero

Testing Your Solution

Use the level-2 input files for this scenario. Each input file contains 250 polynomials encoded as a json string or in the numeric format, with the expected numeric y-value of the polynomial on the following line.

Your solution should read the data files, calculate the result of the polynomial (according to the rules of the exercise) and ensure your value equates to the expected value provided in the file.

Once you have all tests passing AND you are satisfied with the quality of code your have written - **you are ready to submit your solution** - please see *Integration Instructions* for instructions on submitting your solution.

What We're Looking For

In this assessment exercise we are assessing your design and coding skills so your code should be of the highest quality – well designed using OOD/OOP practices (or functional programming techniques), readable, reliable, robust and maintainable. As well as using any techniques that you feel appropriate for high quality and maintainable code you should aim to provide a reasonable level of test coverage for you solution.

When writing unit tests you MAY use mathematical operators other than ADD and SUBTRACT in your test code. .

Third Party Libraries

Please do not use any third party libraries beyond the standard runtime java libraries other than those supplied by us in your solution. For security reasons we will be unable to accept any solution that uses third-party libraries other than those we have supplied (Json api and our Assessment Application implementation libraries - it is **not necessary** to submit these libraries with your solution). If any additional libraries are submitted the solution will be rejected.

If you write any supporting test code please feel free to submit this to us but without the associated third party libraries. We will assess your test code without executing it.

(shown when: *Integration Instructions* button pressed)

Submitting Your Solution

Your solution should be submitted to us via a *private github repository*. Please add our github user **DevAssessmentIDBS** as a collaborator to your repository - this will allow us to access your code in your private repository.

You should include all the source code you write for your solution. If you write any supporting code (e.g. unit test code) - please feel free to submit this code to us as part of the assessment, but without the associated third party libraries (e.g. unit testing libraries), we will assess your test code without executing it. Once submitted to github, please email the url for your repository to us via the email address supplied previously.

Security Considerations

Please do not use any third party libraries other than those supplied within the runtime environment of your selected development language. **For security reasons we will be unable to accept any solution that uses third-party code other than standard libs.**

Final Comments & Thoughts

This is a coding assessment exercise and as such it is necessarily brief. We are looking for a well-designed solution and well written code that is of the highest quality. Do not be afraid to "over design" your solution - the idea is to demonstrate your design and coding skills rather than produce the shortest, simplest solution.

If you have any feedback you wish to share or problems getting started, please contact us on the email used in our previous communications.

Good Luck