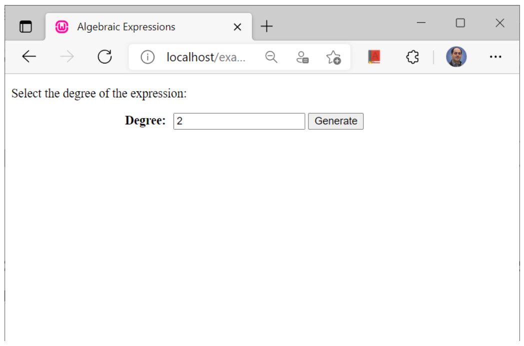
Assignment2 : Using Objects for Random Algebraic Expression Generator

You are asked to refactor (re-write) your assignment1 using class and objects. You are given the solution to assignment1 as attached to this assignment. There are some minor deviations (changes) in assignment 1. That's why you are supposed to modify the provided source codes of assignemnt1 and not your submitted assignement1.

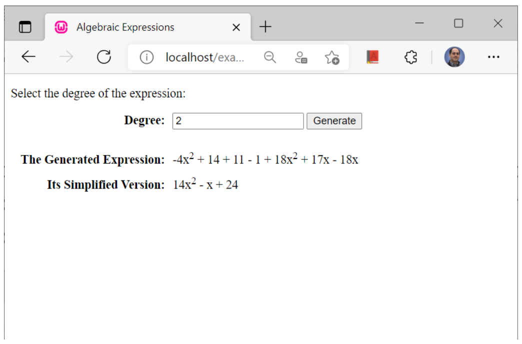
Requirement:

1. You must display a form to the user to enter the degree of the expression. **Only for the first time**, loading the page, the form is displayed alone as you can see in the picture on the left. When the user enters a value for the degree and press Generate button, **from then**, the page looks like the picture on the right. A default value of 2 is displayed in the input box for the degree, but the user can change it to any value greater than 0.
   Read the slides for **Lecture8-Web Forms** to learn how to use an if-else and checking for $_SERVER['REQUEST_METHOD'] to decide if it is the first time the page is loading, or not.



Select the degree of the expression:

Degree: 2 Generate



Select the degree of the expression:

Degree: 2 Generate

The Generated Expression: $-4x^2 + 14 + 11 - 1 + 18x^2 + 17x - 18x$

Its Simplified Version: $14x^2 - x + 24$

2. To display the form, define a function **display_form.** This function has one input parameter **degree**, it does not return any value. It displays the label for **Degree:** and an input box and submit button. The input parameter degree is used to set the value for input (i.e., <input value="$degree" ……. />) . use HEREdoc to print the HTML code.
3. Define a class called **Term**.
   a. This call has only two properties: **coefficient** and **exponent**. This class is going to represent each single term of an expression.
4. Define another class **Expression.**
   a. This class has two const called **coefficient_min** and **coefficient_max.** these two consts define the lower and upper limit range of random values for coefficients of the expression. Set it to -20 and 20 respectively.
   b. This class has three properties: **terms**, **degree,** and **num_of_terms.**
      i. **terms** property is a single-dimension array of **Term** objects. **NOTE:** it is not a two-dimensional array as in assignment1. **It is an array of objects** (i.e., an array of Term objects)
      ii. **degree** and **num_of_terms** are integer variables
   c. This class has 4 methods: **generate(), stringify(), meets_requirements(), simplify().**
      Note, in the code provided to you the name of functions is, for example, generate_expression(), but as we will use the name of an object with the method, it is simpler in the new edition of the code to change the name from generate_expression to generate. Because, if the name of an object is $expr then it is simpler to write $expr->generate() than $expr->generate_expression() (we do not need to repeat expr and expression!)
5. Create a constructor for the class **Expression**. The constructor has 1 input parameter: **degree**. When we create an Expression object, we pass a value for the degree. For example:

   $expr = new Expression(2);   // to create an expression of degree 2

   Inside the constructor, set the initial value for the **degree** property to the input value to the constructor, also set **num_of_terms** based on the same formula given in assignemnt1, also set **terms** property to an empty array. You know that in a constructor of a class, we initialize the properties of an object.

6. The generate() method of the class does not have any input parameter as we did in the assignment1. Because, the degree is a property for the class now, no need to get the degree as an input parameter to the method. Use **$this->degree** to refer to the degree property.
   Inside the do-while loop, first set **$this->terms** to **null**, to ensure the terms array is empty, then for every term you must create an object from **Term** class, set the **coefficient** and **exponent** properties of the object to randomly generated values and then assign this newly created object from the class Term to the array **terms**. Like:

   $this->terms[] = $term; // $term is an object of the class Term. $this->terms is an array, a property of the class Expression

   NOTE: terms is not a two-dimensional array, as we did in assignement1, **terms is an array of objects** in this assignment.
   generate() returns the property terms (the array of objects representing the terms of the expression)
7. **meets_requirements()** does not have any input, because both of inputs that we assigned to it in assignemnt1 are the properties of the class now and we do not need to get them as input. We refer to $degree as **$this->degree**, also **$this->num_of_terms**, and to refer to an exponent of a term we use **$this->terms[$i]->exponent** instead of **$terms[1][$i]**

   in this function, only reference to the variables (degree, num_of_terms, and exponent of a term) changes and the rest of the body of the function remains unchanged.
8. **stringify()** method has no input and returns a string. Variables must be changed. For example, **$terms[0][0]** is going to be **$this->terms[0]->coefficient**, also **$terms[1][0]** is changed to **$this->terms[0]->exponent**. Also **count($terms[0])** is changed to **count($this->terms)**. Apply any other changes required. No need to make stringify_term() function a method of the class.

9. You must write a new function called **simplify** which is a method of the class Expression. This method gets an object of Expression and simplify it. This method is **static.** This method has 1 input which is an object of the class Expression, and it returns an object of the same type (i.e., simplified version of the input expression which is an object of the class Expression).

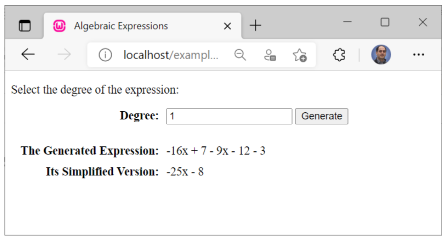
   In this function, first, we create a new object of Expression. Then by a nested loop, we repeat for all degrees to sum up all the coefficients of the terms that have the same exponent. Follow this algorithm:

```
Create a new object of the class Expression
For-loop (repeat $degree times) {  // if the degree is 3, then 3,2,1,0
        Sum = 0
        For-loop(repeat for the length of the array terms of input expression){
                If (exponent of a term is equal to the counter of the outer-loop)
                        Sum += coefficient_of_the_term
        }
        If (sum is not zero)
                Create a new term from the class Term
                Set exponent to the counter of the outer-loop
                Set coefficient to sum
                Assign this new Term object to the array terms of the new simplified Expression object
Set num_of_terms for this new simplified object to the size of its terms array
Return the simplified expression object
```

Assignment2 : Using Objects for Random Algebraic Expression Generator

Screenshots

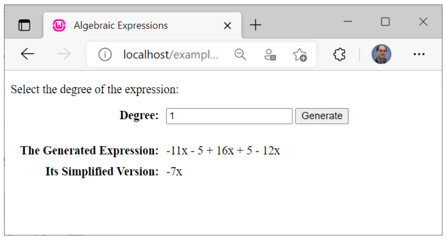
Some screenshots for degree=1:


Select the degree of the expression:

Degree: 1 Generate

The Generated Expression: -16x + 7 - 9x - 12 - 3

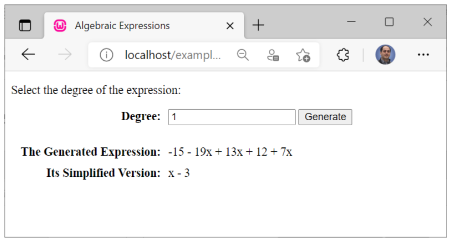
Its Simplified Version: -25x - 8


Select the degree of the expression:

Degree: 1 Generate

The Generated Expression: -11x - 5 + 16x + 5 - 12x

Its Simplified Version: -7x

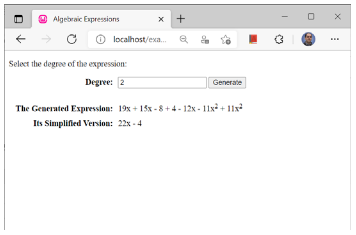

Select the degree of the expression:

Degree: 1 Generate

The Generated Expression: -15 - 19x + 13x + 12 + 7x

Its Simplified Version: x - 3

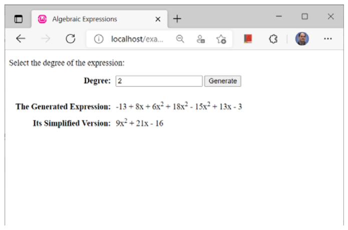
Some screenshots for degree=2:


Select the degree of the expression:

Degree: 2 Generate

The Generated Expression: $19x + 15x - 8 + 4 - 12x - 11x^2 + 11x^2$

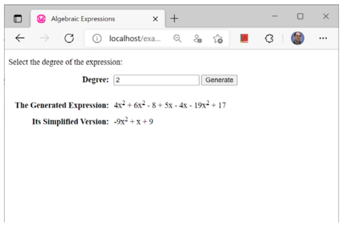
Its Simplified Version: 22x - 4


Select the degree of the expression:

Degree: 2 Generate

The Generated Expression: $-13 + 8x + 6x^2 + 18x^2 - 15x^2 + 13x - 3$

Its Simplified Version: $9x^2 + 21x - 16$

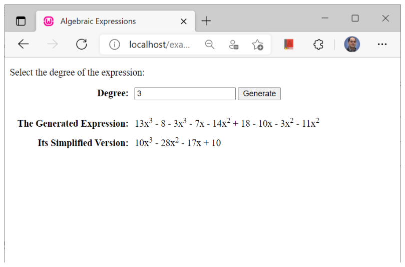

Select the degree of the expression:

Degree: 2 Generate

The Generated Expression: $4x^2 + 6x^2 - 8 + 5x - 4x - 19x^2 + 17$

Its Simplified Version: $-9x^2 + x + 9$

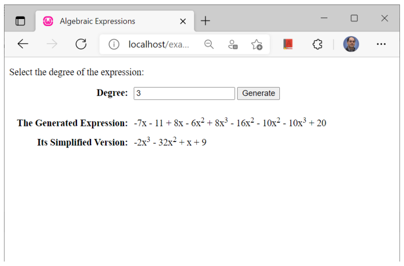
Some screenshots for degree=3:


Select the degree of the expression:

Degree: 3 Generate

The Generated Expression: $13x^3 - 8 - 3x^3 - 7x - 14x^2 + 18 - 10x - 3x^2 - 11x^2$

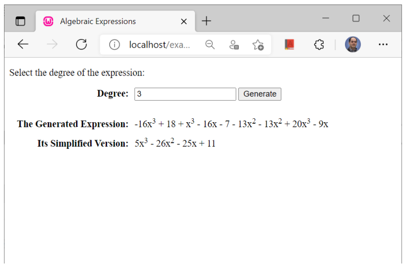
Its Simplified Version: $10x^3 - 28x^2 - 17x + 10$


Select the degree of the expression:

Degree: 3 Generate

The Generated Expression: $-7x - 11 + 8x - 6x^2 + 8x^3 - 16x^2 - 10x^2 - 10x^3 + 20$

Its Simplified Version: $-2x^3 - 32x^2 + x + 9$

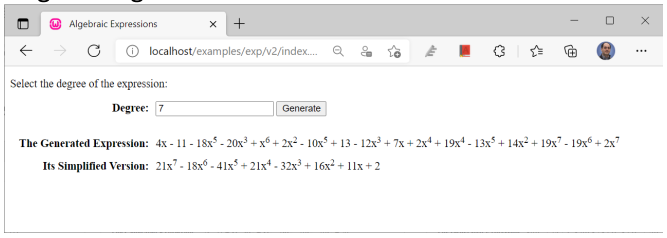

Select the degree of the expression:

Degree: 3 Generate

The Generated Expression: $-16x^3 + 18 + x^3 - 16x - 7 - 13x^2 - 13x^2 + 20x^3 - 9x$

Its Simplified Version: $5x^3 - 26x^2 - 25x + 11$

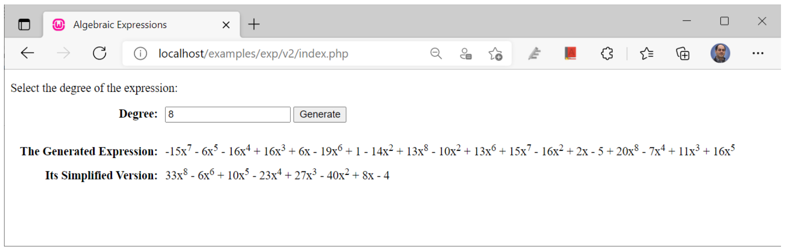
Higher Degrees


Select the degree of the expression:

Degree: 7 Generate

The Generated Expression: $4x - 11 - 18x^5 - 20x^3 + x^6 + 2x^2 - 10x^5 + 13 - 12x^3 + 7x + 2x^4 + 19x^4 - 13x^5 + 14x^2 + 19x^7 - 19x^6 + 2x^7$

Its Simplified Version: $21x^7 - 18x^6 - 41x^5 + 21x^4 - 32x^3 + 16x^2 + 11x + 2$

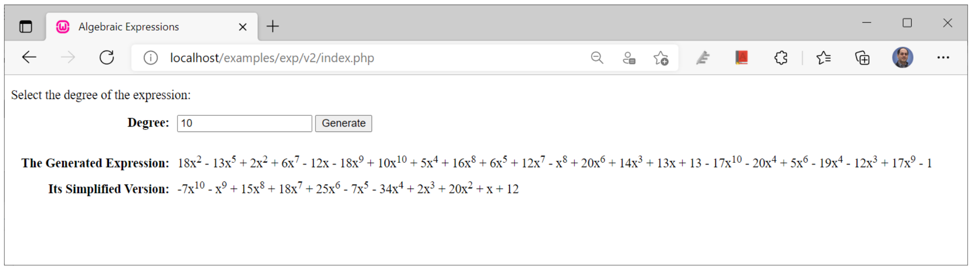

Select the degree of the expression:

Degree: 8 Generate

The Generated Expression: $-15x^7 - 6x^5 - 16x^4 + 16x^3 + 6x - 19x^6 + 1 - 14x^2 + 13x^8 - 10x^2 + 13x^6 + 15x^7 - 16x^2 + 2x - 5 + 20x^8 - 7x^4 + 11x^3 + 16x^5$

Its Simplified Version: $33x^8 - 6x^6 + 10x^5 - 23x^4 + 27x^3 - 40x^2 + 8x - 4$

Assignment2 : Using Objects for Random Algebraic Expression Generator



Algebraic Expressions

localhost/examples/exp/v2/index.php

Select the degree of the expression:

Degree: 10 Generate

The Generated Expression: $18x^2 - 13x^5 + 2x^2 + 6x^7 - 12x - 18x^9 + 10x^{10} + 5x^4 + 16x^8 + 6x^5 + 12x^7 - x^8 + 20x^6 + 14x^3 + 13x + 13 - 17x^{10} - 20x^4 + 5x^6 - 19x^4 - 12x^3 + 17x^9 - 1$

Its Simplified Version: $-7x^{10} - x^9 + 15x^8 + 18x^7 + 25x^6 - 7x^5 - 34x^4 + 2x^3 + 20x^2 + x + 12$

Submission

Please submit the following items to the corresponding assignment folder on Blackboard prior to the deadline:

- A zip/archive folder containing all HTML, PHP source code files. Rename the zip file name as a#_xy.zip where # is assignment number, x is your first name and y is your last name. For example, when I submit my work for assignment 2, I will rename the zip file as a2_SaeedMirjalili.zip
- If you do not know How to zip file(s)?
- Or search Google
- I found this video also useful for zipping and uploading your project

Marking Guideline

Task	Points	Granted
Creating the form using $_SERVER['REQUEST_METHOD']	**15**	
Display_form function	**10**	
Term class	**5**	
Expression Class	**12**	
Constructor of the Expression class	**10**	
Generate function	**10**	
Meets_requirements function	**8**	
Stringify function	**10**	
Simplify static method	**20**	
Total	**100**	**0**