CIS263AA Programming Assignment 12

Assignment Goal:

In this chapter you learned about enumerations and autoboxing. Enumerations are a way to define named constants that define a new data type. Autoboxing makes it easy to convert primitives to object and objects to primitives automatically

Assignment Specifications:

John is a science teacher. He decides he wants a program that allows students to enter an elements symbol and return the element name, atomic number, and atomic weight.

Write an enumeration called **Elements** for at least 10 elements of your choice. The periodic table of elements can easily be found on the web.

Write a program called **DisplayElements** that allows the user to enter a symbol and display its data. If the user enters “All”, call a method that displays all of the elements using a for each loop. Use the name() method to display the symbol. The program should loop until the user enters exit.

Deliverables (what you are to submit):

1. Planning document for Elements
   1. Methods
   2. Data Items
   3. Test Data – This will be your elements
2. Planning document for DisplayElements.
   1. Program Outline
   2. Methods
   3. Data Items
   4. Sample Output
   5. Test Data
3. Your completed project folder in zip format.

CIS263AA Programming Assignment 12 (Elements)

Name: \_\_\_Daniel Cender\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Method):** < What things is the program to do? >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Method Name** | **Return Type (1)** | **Return Value** | **Parameters** |
| There is no main() method in this class. It is an enumeration. | | | | |
| Constructor | Elements() |  |  |  |
| Get Name | getName() | String | name |  |
| Get Atomic Number | getANumber() | int | aNumber |  |
| Get Weight | getWeight() | float | weight |  |

1. Return Type: void, string, char, byte, short, integer, long, double, float, boolean, class object, etc.

**Data Items:** < This is a list of fields (variables, constants, and objects you will need >

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Item** | **Source (1)** | **Access(2)** | **Data Type (3)** | **Identifier** | **Notes** |
| Element name | constant | private | String | name |  |
| Element number | constant | private | int | aNumber |  |
| Element Weight | constant | private | float | weight |  |
| Silver | constant | public | Element | SILVER |  |
| Neon | constant | public | Element | NEON |  |
| Antimony | constant | public | Element | ANTIMONY |  |
| Copper | constant | public | Element | COPPER |  |
| Gallium | constant | public | Element | GALLIUM |  |
| Titanium | constant | public | Element | TITANIUM |  |
| Helium | constant | public | Element | HELIUM |  |
| Krypton | constant | public | Element | KRYPTON |  |
| Iron | constant | public | Element | IRON |  |
| Mercury | constant | public | Element | MERCURY |  |

1. Source: calculated, input, constant, literal, parameter

2. Access: local, private, protected, public

3. Type: string, char, byte, short, integer, long, double, float, etc.

**Test Data:** < How will you prove your program works >

I will test the accessor methods and constructor method by the successful completion and testing of the DisplayElements program.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | **Value 1** | **Value 2** | **Value 3** | **Value 4** | **Value 5** |
| answer | Si | Ne | Sb | Cu | Ga |
| name | Silver | Neon | Antimony | Copper | Gallium |
| aNumber | 14 | 10 | 51 | 29 | 31 |
| weight | 28.085 | 20.1797 | 121.760 | 63.546 | 69.723 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | **Value 6** | **Value 7** | **Value 8** | **Value 9** | **Value 10** |
| answer | Ti | He | Kr | Fe | Hg |
| name | Titanium | Helium | Krypton | Iron | Mercury |
| aNumber | 22 | 2 | 36 | 26 | 80 |
| weight | 47.867 | 4.002 | 83.798 | 55.845 | 200.592 |

Note: You made more or fewer test cases depending on your application.

CIS263AA Programming Assignment 12 (DisplayElements)

Name: \_\_\_\_Daniel Cender\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Program Outline:** < What things is the program to do? >

Ask user for element symbol

Find constant that fits input

Display element properties to the screen

**Methods:** < What methods will you have in your program? >

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Function** | **Method Name** | **Return Type (1)** | **Return Value** | **Parameters** |
| Main | main() | void |  | String[] args |
| Display Element | displayElement() | void |  |  |
| Display All Elements | displayAll() | void |  |  |

1. Return Type: void, string, char, byte, short, integer, long, double, float, boolean, class object, etc.

**Data Items:** < This is a list of fields (variables, constants, and objects you will need >

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Item** | **Source (1)** | **Access(2)** | **Data Type (3)** | **Identifier** | **Notes** |
| Element Enum | constant | local | Elements | element |  |
| Answer | input | local | String | answer |  |
| Input Reader | input | local | BufferedReader | reader |  |

1. Source: calculated, input, constant, literal, parameter

2. Access: local, private, protected, public

3. Type: string, char, byte, short, integer, long, double, float, etc.

**Sample Output:** < What will the user see? >

Please enter an element symbol to get properties.

Available elements are: Si, Ne, Sb, Cu, Ga, Ti, He, Kr, Fe, and Hg.

Input “All” to see all elements.

Enter “exit” to quit program.

XX’s name: XXXXX

XX’s atomic number:99

XX’s atomic weight: 999

(above will be cycled through 10x, once for each element in a new block if All was inputted)

**Test Data:** < How will you prove your program works >

I will record what the output of the input options should be and test them manually for their output

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | **Value 1** | **Value 2** | **Value 3** | **Value 4** | **Value 5** |
| answer | Si | Ne | Sb | Cu | Ga |
| name | Silver | Neon | Antimony | Copper | Gallium |
| aNumber | 14 | 10 | 51 | 29 | 31 |
| weight | 28.085 | 20.1797 | 121.760 | 63.546 | 69.723 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identifier** | **Value 6** | **Value 7** | **Value 8** | **Value 9** | **Value 10** |
| answer | Ti | He | Kr | Fe | Hg |
| name | Titanium | Helium | Krypton | Iron | Mercury |
| aNumber | 22 | 2 | 36 | 26 | 80 |
| weight | 47.867 | 4.002 | 83.798 | 55.845 | 200.592 |

Note: You made more or fewer test cases depending on your application.