All assignments are emailed to cislabs05@gmail.com

Assignment/Lab 2 - OOP

Part 1 - Human Genome Class

Genetics is the study of how traits such as hair color, eye color, and risk for disease are passed ('inherited') from parents to their children. Your genetic code is programmed in Genome (your Genetic code), which is made up of a chemical called DNA (deoxyribonucleic acid) and is stored in almost every cell in your body.

Genomes in a human = 1

Genes in a human genome = 20000

Cells in your body (3 * US Deficit) = 75 to 100 trillion. (In my case it's probably - 200 trillion considering my body mass 😊)

Chromosomes in a human cell = 46.

Object Oriented Programming is an art that allows mapping of relationships between objects (physical or virtual) enabling creation of thinking machines solving complex problems (with usage of AI/ML) that would take humans a long time to solve.

In this course we will marry the two to produce beautiful outcomes. After all, COVID-19 is a crisis, we programmers can possibly help resolve it through understanding the DNA of viruses and finding the DNA that kills the virus. Doing this kind of programming is a branch of Bioinformatics.

In this lab (and the series to follow) we will begin to solve this problem.

Concepts to apply:

Creating a class and several objects (in your personal family) in which we can begin defining the blueprint of our own Genome.

Design a **HumanGenome** class that holds the following information: genome name, number of genes in genome, number of chromosomes, number of cells in your body. Write appropriate methods (constructor, getters and setters and print)

*print() will print the value of all properties using System.out.printf().

Demonstrate the class by writing a program that creates three or more instances of this class (Name the instances using a disney character).

You can populate information in each object using Scanner class. Please do not use any personal information as data in the project.

Submit a class diagram, test runs and code (.java file) with your submission, following java files must be included.

HumanGenome.java - contains properties, constructors and properties as stated above. Also create a print() method printing value of each property as value name pairs in separate lines using printf().

GenomeInput.java -contains input method (instance method) that uses Scanner for data input and returns a HumanGenome instance.

Add a driver (**a main method in GenomeInput class)** that calls the input method 3 times to create three instances of HumanGenome.

Possible input values for HumanGenome properties

Name of property	value1	value2	value3
GenomeName	“Human Bob”	“Human Sally”	“Human Sri”
NumberofGenes	20000	20000	20000
NumberofChromosomes	46	46	46
NumberofCells (Trillions)	76	120	75

Please create a zip file and submit a single attachment for this part.

Part 2 - Create a Nucleic acid

For our purposes, this is a smallest structural unit in DNA and RNA used for creating DNA or RNA.

Create a class called **NucleicAcid** with following properties

1. Name - String
2. ChemicalFormula - String
3. Molarmass - float (and units can be added as a String (within printf())
4. Density - float (and units can be added as a String (within printf())

Write appropriate methods (constructor, getters, setters and print). Also create a print() method printing value of each property as value name pairs in separate lines using System.out.printf().

Create a Driver that instantiates following Nucleic Acids demonstrating the classes.

Cytosine
- Chemical formula - C4H5N3O
- Molar mass - 111.10 g/mol
- Density - 1.55 g/cm3

Adenine
- Chemical formula C5H5N5
- Molar mass 135.13 g/mol
- Density 1.6 g/cm3

Guanine
- Chemical formula C5H5N5O
- Molar mass 151.13 g/mol
- Density 2.200 g/cm3

Thymine
- Chemical formula C5H6N2O2
- Molar mass 126.115 g·mol−1
- Density 1.223 g cm−3

Uracil
- Chemical formula C4H4N2O2
- Molar mass 112.08676 g/mol
- Density 1.32 g/cm3

Submit a class diagram, test runs and code (.java file) with your submission, following java files must be included.

NucleicAcid.java - contains properties, constructors and properties as stated above. Also create a print() method printing value of each property as value name pairs in separate lines using printf().

GenoNucleicAcid.java (similar to the Driver program) - contains an input method (instance method) that uses **Scanner** for data input and returns a NucleicAcid instance. (Use the data stated above for instantiating Nucleic Acids).

Please create a zip file and submit a single attachment for this part.

****

Java coding standards

When writing a class:

- Class name should start with an uppercase letter.
- Class should always be public
- Must include a default constructor.
- All instance variables (properties) must be private
- All methods must be public or protected
- Class anatomy - instance variables, static variables, constructors, getter and setters, instance methods, static methods.
    - --instance variables
    - --static variables
    - --constructors
    - --instance methods
    - ----getters/setters
    - ----business methods (calculations)
    - ----print()
    - --static methods - usually update static variables [will not be used to update values of
    - instance variables] -- because static methods can be used without creating an object.
- Driver (a class with a main() in it) should be in its own class.
- Every class must have a print method that can print the values of instance variables - must System.out.printf.
- One class per .java file

****