|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| **Programming Assignment 2**  **Date Posted: February 25 Preliminary Class Diagram: March 3 Class Outline: March 10 Functionality Outline: March 17 Final Project Due: Thursday, March 31**  **Programming Assignment 2** | | | | |
| **Background Information** | | | | |
|  | In 1900 after almost forty years in obscurity the work of Gregor Mendel was rediscovered by Hugo de Vries and Carl Correns. In 1902 Theodor Boveri, a German geneticist, and Walter Sutton, an American geneticist, independently established that genes are "carried" on chromosomes. Each species of plant or animal has a certain number of chromosomes arranged in pairs in each of their cells. One chromosome of each pair is inherited from the father and one from the mother.  Shortly after that two British geneticists, William Bateson and Reginald Punnett of St. John’s College, Cambridge, discovered that certain traits always seemed to be inherited together. These traits, which seem in contradiction to the Mendelian *Law of Independent Assortment*, are called **linked** traits. They are almost always inherited together because they reside on the same chromosome. The key words here are "almost always inherited together". It was later discovered that through a process called **crossing over** certain segments of chromosomes can swap places, that is a segment of genes residing on one chromosome of a pair can swap places with the same segment on its' chromosome pair. It was also discovered that the frequency of such crossing over was related to the distance two genes were separated on the chromosome which they shared.   Bateson, by the way, was the first to use the term ***genetics*** (from the Greek ***genno***, http://www.cs.uah.edu/~rcoleman/CS307/ProgAssign/Images/GreekGenno.jpg meaning to give birth) to describe the study of heredity and biological inheritance. | | | |
| **Your Assignment** | | | | |
|  | You are to write a simulation program to demonstrate the processes of inheritance incorporating the Mendelian laws of genetics and later information on chromosomal structure and gene linking.  The design and implementation of this program **must** follow object oriented design principles. Opportunity will be provided for you to question the customer's representative (the instructor) in class for more details.  All requirements from programming assignment 1 are to be followed unless modified in the following list of requirements for this simulation.   1. The simulation shall be able to define two "parent" organisms and their genotype for a test run. Data defining an organism shall consist of a name (given as a *genus species*, such as *Drosophila Melanogaster*), the number of chromosomes being simulated, the number and type of genes on each chromosome in the genotype, and data defining each gene. All data defining the two organisms shall be read from a data file which will be provided. The format of the data file is given below. Two sample data files will be provided. The application shall be capable of running with either data file or any other data file in the same basic format. A new data parser class will also be provided. The new data parser class will follow the Singleton Design Pattern. 2. The simulation shall represent the genotype of an organism as any number of instances of a chromosome object, each containing any number of gene objects. A Chromosome will contain, at a minimum, the following information.    1. A collection of the genes in the chromosome    2. Data specifying which allele of each gene pair is on each strand of the chromosome.   A Gene will contain, at a minimum, the following information.   * 1. A brief description of the trait, e.g. "Eye Color".   2. The specific phenotypes (displayed traits) represented by each allele of this gene, e.g. "Red Eyes" and "White Eyes".   3. An indicator as to which trait is the dominant trait.   4. A character to represent each allele. Note: A capitol letter shall be used to represent a dominant trait and its' corresponding lower case letter to represent the recessive trait of the pair.   5. A double specifying the percentage chance that this gene can cross over, i.e. swap places with its mate from the other chromosome in the pair.  1. After reading all data defining two parent organisms the simulation shall then query the user for the number of offspring to generate. This can be in the range of 1 to 1000. A Mendelian cross between the two organisms shall then be performed taking into account the fact that some traits are linked and the chance of a crossover occurring. The results shall be printed on the screen. The format of the output shall follow the outline given below. | | | |
|  | **Structural Requirements**   1. Genes shall be represented following a variation of the Flyweight Design Pattern. In this there shall be one instance of a "master gene" for each gene type. This object shall hold all the information defining a particular gene. To represent the genes in an organism there shall be a simple "gene" class which has a reference to the master gene but only contains the specific allele characters for that instance of the gene. 2. The creation of instances of "master genes" and "genes" shall be encapsulated in a Gene Factory class which is a simple factory.\* 3. The creation of instances of chromosomes shall be encapsulated in a Chromosome Factory class which is a simple factory.\* The Chromosome Factory shall use the Gene Factory to create genes for chromosomes. 4. The creation of instances of organisms shall be encapsulated in an Organism Factory class which is a simple factory.\* The Organism Factory shall use the Chromosome Factory to create chromosomes for organisms. The organism factory shall contain a function which returns an instance of an organism with fully defined genotype. 5. All of the factory classes shall be created following the Singleton Design Pattern.   \*A simple factory is not one of the design patterns, but it has some of the characteristics of the Factory Method and the Abstract Factory Design Patterns. We will discuss this in class. | | | |
| **Deliverables** | | | | |
|  | These products as specified below shall be delivered electronically via e-mail to the instructor.  **Preliminary Class Diagram --**The class diagram shall be drawn using standard UML notation and shall show all of the classes to be implemented in the software and their relationships (dependencies, associations, generalizations, realizations, etc.) The PCD shall be submitted for instructor approval NLT (Not Later Than) Thursday, March 3.  **Class Outline --**The class outline shall list all proposed variables and functions in each proposed class with a brief description of what each does. The class outline shall be submitted for instructor approval NLT (Not Later Than) Thursday, March 10.  **Functionality Outline --**The functionality outline shall be an outline which will show the step-by-step functionality of the program. This should be taken out to a fair amount of detail. The functionality outline shall be submitted for instructor approval NLT (Not Later Than) Thursday, March 17.  **Final Project --**The entire software project (compatible with Microsoft Visual Studio 2010 or 2012) shall be compressed into a zip file and submitted for instructor approval NLT Thursday, March 31. **Just turning in your source files is not acceptable.** | | | |
| **We will have several class periods in which we will meet as a team to discuss and plan this project. We will be doing a lot of brainstorming and planning together, but remember that each person is responsibility for implementing the final design on their own.** | | | | |
| **To download a sample executable as well as the sample data files and a parser for the data files click**[**here**](http://www.cs.uah.edu/~rcoleman/CS307/downloads/index.html)**.** | | | | |
| **Input File Format** | | | | |
|  | A data parser class (GeneticsSimDataParser.h and .cpp) will be provided for reading, parsing, and providing data from the data files. Two data files will be provided by the instructor they will be in a modified XML format. Below is a sample of the type of data file that will be used by the simulation.  <!-- Sample data file for use in the Mendelian Genetics Simulation Program 2 -->  <!-- Note: This file is not fully compatible with XML standards, but close enough -->  <!-- for the purposes of this project -->  <MENDELIAN\_GENETICS\_SIM>  <ORGANISM>  <GENUS>  Pisum  </GENUS>  <SPECIES>  Sativum  </SPECIES>  <COMMON\_NAME>  Pea Plant  <COMMON\_NAME>  <CHROMOSOME\_COUNT>  2  </CHROMOSOME\_COUNT>  </ORGANISM>  <GENES>  <GENE>  <GENE\_TRAIT>  Plant Stature  </GENE\_TRAIT>  <DOMINANT\_ALLELE>  Tall  </DOMINANT\_ALLELE>  <DOMINANT\_SYMBOL>  T  </DOMINANT\_SYMBOL>  <RECESSIVE\_ALLELE>  Dwarf  </RECESSIVE\_ALLELE>  <RECESSIVE\_SYMBOL>  t  </RECESSIVE\_SYMBOL>  <CROSSOVER\_CHANCE>  5.2  </CROSSOVER\_CHANCE>  </GENE>  <GENE>  <GENE\_TRAIT>  Seed Texture  </GENE\_TRAIT>  <DOMINANT\_ALLELE>  Wrinkled  </DOMINANT\_ALLELE>  <DOMINANT\_SYMBOL>  W  </DOMINANT\_SYMBOL>  <RECESSIVE\_ALLELE>  Smooth  </RECESSIVE\_ALLELE>  <RECESSIVE\_SYMBOL>  w  </RECESSIVE\_SYMBOL>  <CROSSOVER\_CHANCE>  4.3  </CROSSOVER\_CHANCE>  </GENE>  <GENE>  <GENE\_TRAIT>  Seed Color  </GENE\_TRAIT>  <DOMINANT\_ALLELE>  Green  </DOMINANT\_ALLELE>  <DOMINANT\_SYMBOL>  S  </DOMINANT\_SYMBOL>  <RECESSIVE\_ALLELE>  Yellow  </RECESSIVE\_ALLELE>  <RECESSIVE\_SYMBOL>  s  </RECESSIVE\_SYMBOL>  <CROSSOVER\_CHANCE>  4.25  </CROSSOVER\_CHANCE>  </GENE>  <GENE>  <GENE\_TRAIT>  Flower Color  </GENE\_TRAIT>  <DOMINANT\_ALLELE>  Purple  </DOMINANT\_ALLELE>  <DOMINANT\_SYMBOL>  C  </DOMINANT\_SYMBOL>  <RECESSIVE\_ALLELE>  White  </RECESSIVE\_ALLELE>  <RECESSIVE\_SYMBOL>  c  </RECESSIVE\_SYMBOL>  <CROSSOVER\_CHANCE>  5.75  </CROSSOVER\_CHANCE>  </GENE>  </GENES>  <PARENTS>  <PARENT>  <CHROMOSOME>  <STRAND1>  T C  </STRAND1>  <STRAND2>  t c  </STRAND2>  </CHROMOSOME>  <CHROMOSOME>  <STRAND1>  W S  </STRAND1>  <STRAND2>  w s  </STRAND2>  </CHROMOSOME>  </PARENT>  <PARENT>  <CHROMOSOME>  <STRAND1>  T C  </STRAND1>  <STRAND2>  t c  </STRAND2>  </CHROMOSOME>  <CHROMOSOME>  <STRAND1>  W S  </STRAND1>  <STRAND2>  w s  </STRAND2>  </CHROMOSOME>  </PARENT>  </PARENTS>  <MENDELIAN\_GENETICS\_SIM> | | | |
| **Output Format** | | | | |
|  | The output printed on the screen shall contain the results of an experimental run in the following format:  Master Genes:  Trait Name: Plant Stature  Dominant Name: Tall(T)  Recessive Name: Dwarf(t)  Chance of crossover: 5.2  Trait Name: Seed Texture  Dominant Name: Wrinkled(W)  Recessive Name: Smooth(w)  Chance of crossover: 4.3  Trait Name: Seed Color  Dominant Name: Green(S)  Recessive Name: Yellow(s)  Chance of crossover: 4.25  Trait Name: Flower Color  Dominant Name: Purple(C)  Recessive Name: White(c)  Chance of crossover: 5.75  Sim parent 1  Organism genus-species: Pisum Sativum  Chromosomes:  Chromosome 1  Gene Type: Plant Stature  Allele 1: Tall(T)  Allele 2: Dwarf(t)  Gene Type: Flower Color  Allele 1: Purple(C)  Allele 2: White(c)  Chromosome 2  Gene Type: Seed Texture  Allele 1: Wrinkled(W)  Allele 2: Smooth(w)  Gene Type: Seed Color  Allele 1: Green(S)  Allele 2: Yellow(s)  Sim parent 2  Organism genus-species: Pisum Sativum  Chromosomes:  Chromosome 1  Gene Type: Plant Stature  Allele 1: Tall(T)  Allele 2: Dwarf(t)  Gene Type: Flower Color  Allele 1: Purple(C)  Allele 2: White(c)  Chromosome 2  Gene Type: Seed Texture  Allele 1: Wrinkled(W)  Allele 2: Smooth(w)  Gene Type: Seed Color  Allele 1: Green(S)  Allele 2: Yellow(s)  How many offspring do you want to generate? (Type the number then press Enter)  -->50  ======================= Results of this Run =======================  Gene: Plant Stature  11 homozygous dominant (Tall TT)  23 heterozygous dominant (Tall Tt)  16 homozygous recessive (Dwarf tt)  Gene: Flower Color  8 homozygous dominant (Purple CC)  27 heterozygous dominant (Purple Cc)  15 homozygous recessive (White cc)  Gene: Seed Texture  13 homozygous dominant (Wrinkled WW)  25 heterozygous dominant (Wrinkled Ww)  12 homozygous recessive (Smooth ww)  Gene: Seed Color  12 homozygous dominant (Green SS)  25 heterozygous dominant (Green Ss)  13 homozygous recessive (Yellow ss)  A total of 14 offspring had at least one crossover gene. | | | |