COMP1006/1406 – Fall 2016

Submit a single file called `assignment5.zip` to cuLearn.
The assignment is out of 30 marks.

1: Reading [5 marks]

Download the **Space Invaders** code (from cuLearn). This code is written by Kevin Glass and is part of a tutorial on accelerated graphics (in Java) that he created.
(http://www.cokeandcode.com/info/tut2d.html)

A) Read through the different classes in the code. Give a **brief** description of each class that is provided. (A high level description in **less** than five lines of writing for each.)

B) Compile and run the `Game` program. Briefly describe two (2) changes to the code that would make for a more interesting game. (Things that you could do.)

You will be using this code in future tutorials and assignments. (You will be extending and modifying the code.) Take some time now to make sure you know what this code is and how it works.

Write your answers in a single plain text file called `Space.txt`.

Include your `Space.txt` file in your `assignment5.zip`.

People, Bugs and Disease

The remainder of this assignment involves working with classes that will interact with each other. These involve people, bugs and diseases.

2: Person Class [5 marks]

Complete the provided `Person` class. Add appropriate getter and setter methods. Complete the `interact(Person)` method.

When a person transmits a disease to another person, a new copy of the same disease should be transmitted (same name, severity, chance of transmission). Make sure that you create a new Disease object. (why?)

Put your `Person.java` file in your `assignment5.zip` file.

3: Diseases [10 marks]

Create two classes: `Zika` and `Plague`. Each of these classes must extend the `Disease` class. Each of these classes must define (override) both abstract methods inherited from Disease. For each, when creating objects, the name passed in should just be the name of the disease.

- `Zika` class: The severity should be an integer between 0 and 10 (inclusive).

```
public void impact(Person p)
   // Precondition: none
   // Postconditions: none
   // Side Effects:
   //  (1) the person's health will be decreased by the value
   //      of the severity of the disease
   //  (2) The disease's severity will be reduced by 1 (until it reaches zero)
   //  (3) The disease's chance of transmission is reduced by 10%
   //      (i.e. 0.9 * current chance of transmission)

public void treat()
   // Pre and postconditions: none
   // Side Effects: Both the severity and chance of transmission are
   //               reduced by a factor of 2. (i.e. each is replaced by 1/2
   //               of the current value, using integer division for severity)
```

- `Plague` class: The severity should be an integer between 0 and 100 (inclusive).

```
public void impact(Person p)
   // Pre and postconditions: none
   // Side Effects:
   //  (1) the person's health will be decreased by 5 if the severity
   //      if larger than 50 and decreased by 2 if the severity is
   //      less than or equal to 50

public void treat()
   // Pre and postconditions: none
   // Side Effects: Severity is reduced by a factor of 3 (i.e. it is replaced by 1/3
   //               of the current value, using integer division)
   //               Chance of transmission remains the same.
```

Put your `Zika.java` and `Plague.java` files in your `assignment5.zip` file.

Note: You may want to create more disease subclasses when testing your code.

4: Insects [10 marks]

Create two classes: `Mosquito` and `Tick`. Each of these classes will extend the `Insect` class. Each class must have the following methods/constructors defined.

- `Mosquito` class: A mosquito's health is an integer between 0 (dead) and 100 (very active).

```
public Mosquito(String name)
   // create a mosquito with no disease and health 100

public Mosquito(String name, int health, Disease disease)
   // create a mosquito that has the given health and disease
   // (disease might be null; meaning no disease)

public void bite(Person p)
   // Pre and postconditions: none
   // Side Effects:
   //  (1) The person's health will be decreased by 1
   //  (2) The mosquito's health will be decreased by 1
   //  (3) If the mosquito's health (before being reduced by 1) is
   //      greater than 50 and it has a disease, it transmits the disease
   //      to the person (if the person doesn't already have this disease)
   //
   //      If the mosquito's health is less than or equal to 50 and
   //      has a disease, it gives the disease to the person
   //      (if the person doesn't already have it), but the disease it
   //      gives has 1/2 the severity of the disease the mosquito has.
```

- `Tick` class: A tick's health is an integer between 0 (dead) and 100 (very active).

```
public Tick(String name)
   // create a tick with no disease and health 100

public Tick(String name, int health, Disease disease)
   // create a tick that has the given health and disease
   // (disease might be null)

public void bite(Person p)
   // Pre and postconditions: none
   // Side Effects:
   //  (1) The person's health will be decreased by 5
   //  (2) The tick's health will be decreased by 1
   //  (3) If the Tick has a disease it gives it to the person.
   //
   //      If the person already has this disease, the person's disease should
   //      be replaced with the same disease having the max severity and
   //      max chance of transmission (of the person's existing disease and
   //      the one that the tick just gave)
   //
```

```
//      Example:
//       if a tick has the plague with severity 99 and chance of
//       transmission 0.3, and the person it is biting already
//       has the plague with severity 80 and chance of transmission
//       of 0.8, then the person will end up with the plague with
//       severity 99 and chance of transmission 0.8.
//
```

Put your `Mosquito.java` and `Tick.java` files in your `assignment5.zip` file.

Submission Recap

A complete assignment will consist of a single file (`assignment5.zip`) with the following five files included: `Space.txt`, `Zika.java`, `Plague.java`, `Mosquito.java`, and `Tick.java`.