**Purpose/Overview**

The program runs a simulation of a world inhabited by three different types of creatures.

**Requirements**

Three types of creatures

The creatures must be able to reproduce

At least one of the creatures must die of old age

The program should support stepping forward one or more generations.

The program should support addition and removal of creatures.

**Classes**

Grid:

* Stores the world map.
* Data members:
  + vector< vector<Creature\*> > grid: stores the map
  + int gen: stores the generation number
* Methods:
  + void display(): prints the map
  + void next(): advances the world a generation
  + void reset(): initializes the world
  + void set(int x, int y, Creature\* crit): adds a creature at x,y

Creature:

* Abstract class.
* Data members
  + int age: age of the creature
  + int maxAge: age at which the creature must die
  + int starve: time since the creature last ate
  + int maxStarve: the maximum a creature can last without eating
  + string sym: symbol representing the creature
* Methods:
  + void move(int, int, int, int, int&, int&): provides the coordinates that the creature moves to
  + void move(int, int, int, int, int&, int&): provides the coordinates that the creature’s child goes to.

Derived classes:

Blank:

Represents a blank cell.

Phytoplankton:

Maximum age of 50, never starves. Does not move or reproduce, but appears at any cell that has been unoccupied for 4 generations. Represented by ‘-’

Zooplankton:

Maximum age of 20, starves after 5 moves. Moves by one cell in any random direction per generation. Produces a child at the right cell at ages 7 and 14.Represented by ‘+’.

Krill:

Maximum age of 15, starves after 4 moves. Moves by one cell in any random direction per generation. Produces a child in any one randomly chosen adjacent cell at ages 5 and 10.

**Global Data/Functions**

No global variables

The main() function is in a separate file main.cpp

**High-level architecture**

The main method creates a Grid object, and accepts user input to manipulate it.

The Grid object contains a 2d vector of pointers to various Creatures, which represent either blank spaces or creatures of different types.

To advance a generation, the program examines every cell one by one.

* Blank cells that have been empty for the last four generations are replaced with Phytoplankton
* If a creature ages beyond its life expectancy or if it starves, it is removed from the map.
* If a creature can move, its new coordinates are calculated and it is moved. If it encounters a creature lower than it in the food chain, it eats it; if it encounters a higher creature, it gets eaten. If it sees a similar creature, it does not move at all.
* Similarly, if the creature can reproduce, the coordinates of its child are calculated and the child is placed in the required cell, to eat or to be eaten depending on the type of creature present.

**User Interface**

The program uses a command-line interface. The initial area of the world are provided via command-line arguments ‘–h=’ and ‘–w=’. The user chooses among the numeric options, and then enters required data via keyboard. Output is displayed as text in the command-prompt.

**Test cases**

The program is tested by spawning creatures at arbitrary points and tracking their movements across the map.