

Module C

Goals-

Design, write, compile, and test a program that uses class inheritance

This module builds on the previous module. You will receive feedback as soon as possible.

Save a copy of the previous module so you have a known place to start in case of catastrophe! Design your changes. Review the design. Then, and only then, start coding!

You will build on the previous version. Make sure you fixed any errors or problems!

You will modify your program to add a specific Critter: Ant. You will design a (simple) class hierarchy to plan the inheritance. Ant will inherit all data and function members of Critter. In addition, it will have a function Breed() and a member variable Count.

You will need to change your array(s) to contain pointers to Ant. **You will not change any other code in your main program.**

Breed() will create a new Ant. It will check the value of count. If count equals 3 then when the Ant moves it leaves a new Ant in the cell it left. It will reset count to 0. Will the Breed() function be public or private?

Think through the algorithm(s). Are you missing anything? If not, it's time to start coding. I have to ask, you weren't coding already, were you? 😊

HINT: You will have new criteria for ending your program. What is it?

NOTE: It may seem pointless to replace Critter with an Ant. There are more critters coming in later modules! We need to make sure it works with a subclass first.

Grading

Modules will be graded S/U. As part of the cumulative modular program (Cump) this module will contribute to that numeric score. These criteria are given for reference.

Programming style- 10%

Create the class hierarchy- 20%

The count data member is used correctly- 10%

Breed() correctly creates the new Ant object- 30%

The 'parent' Ant moves correctly, leaving the new Ant in the original cell - 20%

Your program terminates correctly- 10%