

# PROGRAM THREE – MAGICAL CREATURE ZOO



## CONTENTS

Assignment & Due Dates .....	2
Description:.....	2
What's New in Program 3 .....	2
Multiple Files.....	2
Specifications .....	3
Structures.....	3
Main (define the main function in creatures.cpp) .....	3
Extra Credit Option .....	4
enterCreatures function .....	4
Option 1 – load my creatures from a file .....	5
Option 2 – Enter One Creature Manually .....	5
convertToFloat function .....	6
printCreatures function .....	7
Extra Credit Option .....	8
printStatistics function.....	8
saveCreaturesToFile function .....	9
Extra credit – delete creature function.....	10
How do I compile a program with multiple files? .....	12
What is a makefile?.....	12
What is a Batch File? .....	13

## ASSIGNMENT & DUE DATES

- **Assignment Date:** Monday, October 23, 2017
- **Due Date:** Monday, November 13, 2017

## DESCRIPTION:

Your assignment is to write a program for the Magical Creatures Zookeeper named Hagrid. Hagrid can house up to 100 magical creatures. Information about each creature is stored and manipulated in this program. Hagrid should be able to load magical creature information from any file he or she chooses, add creatures manually, print creature information either to a file or to the screen, or print a cost analysis of each creature and then the total cost to house and take care of these creatures. For 5 points extra credit, you can allow Hagrid to delete creatures as well as one of the main menu options. For another 5 points of extra credit, you can implement word-wrapping when printing out the creature description.

## WHAT'S NEW IN PROGRAM 3

- Structures
- Functions
- Multiple Files

## MULTIPLE FILES

You need to implement this program in three separate files (Creatures.h, Creatures.cpp, & Functions.cpp). You also need to provide me with a text file containing FIVE creatures that YOU come up with (don't use my examples!!!) Below are the list of files that should be included in your zipped submission:

- **Creatures.h** – header file which contains all #includes for the program, structure declarations, and function prototypes
- **Creatures.cpp** – contains the main function
- **Functions.cpp** – contains all the other functions
- **Makefile** – provided for you [if you do not have Windows you will not be able to use this]
- **runProgram.bat** – provided for you [if you do not have Windows you will not be able to use this]
- **my\_creatures.txt** – text file containing data on FIVE creatures that YOU come up with....all data should be separated by hashtags '#'



## SPECIFICATIONS

### STRUCTURES

You will need to create two structures. One is called **Cost**. Cost will have the following members:

- The number of hours it takes to take care of a specific creature for one week
- The cost (per hour) of taking care of this creature.
- The cost of food to feed this creature for one week.
- The cost of materials/supplies (grooming, medical) for this creature for one week

The second structure is called **Creatures**. Creatures will have the following members:

- The name of the creature
- The description of the creature
- The average length of the creature (in feet)
- The average height of the creature (in feet)
- The location of the creature (example: its origin or where they are commonly found)
- If the creature is dangerous (Boolean variable)
- A variable to hold the Cost structure members

### MAIN (DEFINE THE MAIN FUNCTION IN CREATURES.CPP)

You will need to create an array of 100 elements of the **Creatures** data type. You will also need to create a variable that will keep up with the current number of creatures (because there won't always be 100 creatures in Hagrid's zoo).

The main function will display a menu of options like the screen capture below:

```
what would you like to do?
    1. Enter new magical creatures or load from file.
    2. Print all creatures.
    3. Print statistics on creature cost.
    4. End program.
    Enter 1, 2, 3, or 4.
CHOICE:
```

If you decide you want 5 points extra credit, you can also add one more menu option, to "Delete a magical creature". If you do this, it should be option #2 like in the screen capture below.

```
what would you like to do?
    1. Enter new magical creatures or load from file.
    2. Delete a magical creature.
    3. Print all creatures.
    4. Print statistics on creature cost.
    5. End program.
    Enter 1, 2, 3, 4, or 5.
CHOICE:
```

Make sure you always validate all user choices in the whole program before proceeding to do what the user wants. **Each menu choice will call a function that you will create.**

If the user chooses option 1, then your program will call the **enterCreatures** function. If the user chooses option 2, then your program will call the **printCreatures** function. If the user chooses option 3, then your program will call the **printStatistics** function. If the user chooses option 4, then your program will ask the user if they wish to save their creature list to a file. If they choose yes, then your program should call the **saveCreaturesToFile** function and then end. If they choose no, then your program should just end. Do not call the saveCreaturesToFile function unless the user wants to do this!

```

what would you like to do?
    1. Enter new magical creatures or load from file.
    2. Print all creatures.
    3. Print statistics on creature cost.
    4. End program.
    Enter 1, 2, 3, or 4.
CHOICE: 4

Would you like to save your creature list to a file? (y or n) y

```

OR

```

CHOICE: 4

Would you like to save your creature list to a file? (y or n) n

GOODBYE!

C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE

```

**ALL FUNCTIONS OTHER THAN main SHOULD BE DEFINED IN Functions.cpp**

### EXTRA CREDIT OPTION

If you decide you want 5 points extra credit, you can also add one more menu option, to “Delete a magical creature”. If you do this, it should be option #2 like in the screen capture below.

```

what would you like to do?
    1. Enter new magical creatures or load from file.
    2. Delete a magical creature.
    3. Print all creatures.
    4. Print statistics on creature cost.
    5. End program.
    Enter 1, 2, 3, 4, or 5.
CHOICE:

```

Also, this changes which function you call depending on which option they chose. Now, if the user chooses option 2, then your program will call the `deleteCreature` function. Option 3 will call the `printCreatures` function and so forth.

### ENTERCREATURES FUNCTION

The `enterCreatures()` function takes two parameters: the number of creatures currently loaded in the `Creatures` array and the `Creatures` array. **The function will return the new number of creatures.** Before trying to add any creatures, this function should first check to make sure the number of creatures is not already 100. Because if it is, then your program should not add any creatures, but should instead tell the user that their zoo is at full capacity and that they are not able to add creatures. Then the function should end.

If the zoo is not at full capacity then your program will display a menu asking the user if they would like to do one of the following:

```

what do you want to do?
    1. Load my creatures from a file.
    2. Enter one creature manually.
    Choose 1 or 2.
CHOICE:

```

Make sure to validate the user's choice.

---

### OPTION 1 – LOAD MY CREATURES FROM A FILE

If the user chooses option 1, then ask the user what the name of the file is that they would like to load the creatures from.

```
What is the name of the file with your list of creatures? (ex: filename.txt)
FILENAME:
```

Try to open the file. You should print out a message that the file couldn't be opened if it can't be found. Don't continue to try to read from the file or say how many creatures were read from the file if the file couldn't be opened. Below is an example where the file couldn't be opened:

```
1. Load my creatures from a file.
2. Enter one creature manually.
Choose 1 or 2.
CHOICE: 1

What is the name of the file with your list of creatures? (ex: filename.txt)
FILENAME: blah.txt

blah.txt does not exist or is corrupt. Sorry. Can't load creatures.

What would you like to do?
1. Enter new magical creatures or load from file.
```

However if the file could be opened, read each creature from the file and place them in the Creatures array, making sure that you increment the current number of creatures each time a creature is added. When you are reading from the file, everything read in will have to be read in as a string. Some of the Creature members are strings, so that won't be a problem. However, some of the Creature members are floats. So, this function will have to call the `convertToFloat()` function (provided later) in order to convert the string to a float and then placed in the Creatures array.

Print out a verification to tell Hagrid how many creatures have been read from their chosen file.

```
What is the name of the file with your list of creatures? (ex: filename.txt)
FILENAME: my_creatures.txt

4 creatures from my_creatures.txt have been added to the program.
```

---

### OPTION 2 – ENTER ONE CREATURE MANUALLY

If the user chooses option 2, then ask the user for all the creature's details, making sure you place each bit of information in the correct place in the `Creatures` array. The screen capture below shows each piece of information you should ask for. All prompts are in black with no highlighting. All sample user input is highlighted in yellow. The words circled in red indicate this is a variable that will be printed out.



NAME: Unicorn

DESCRIPTION: Big white horse with a horn. Can teleport and only allows the purist of people to approach it. Deadly when angry. Can be calmed down with skittles.

AVERAGE LENGTH (in feet): 16

AVERAGE HEIGHT (in feet): 7

LOCATION: England

IS IT A DANGEROUS CREATURE? (y or n): y

How many hours do you spend caring for the Unicorn?

NUM HOURS: 15

What is the cost per hour for caring for the Unicorn?

COST PER HOUR: 100

How much money do you spend on food for the Unicorn?

FOOD COST: 20

How much money do you spend on grooming and medical supplies for the Unicorn?

SUPPLY COST: 30.15

The Unicorn has been added.

Then, increment the number of creatures by one.



## CONVERTTOFLOAT FUNCTION

The `convertToFloat()` function will need to be used when reading data from a file and inputting the data into the Creatures array (so this function will be called from the `enterCreatures` function). I am providing this function for you. You just need to insert it into your program so you can use it.

```
#include <sstream> //this should be added in Creatures.h

//The following function should be added in Functions.cpp
float convertToFloat(string s)
{
    istringstream mystring(s);
    float myfloat;
    if (!(mystring >> myfloat))
        myfloat = 0;
    return myfloat;
}
```

## PRINTCREATURES FUNCTION

The `printCreatures()` function is a void function and contains the following parameters: number of creatures currently in the Creatures array and the Creatures array.

Check if there are any creatures. If not, then print "THERE ARE NO CREATURES AT YOUR ZOO!".

```
-----  
THERE ARE NO CREATURES AT YOUR ZOO!
```

Otherwise, print out all the creatures array to the screen in the following format:

```
-----  
CREATURE 1:  
Name:      Beholder  
Description:  
Giant center eye and twelve eye stalks above it. It is a flying eyeball. Mouth  
full of razor sharp teeth. Eye stalks shoot various beams of magical death-dea  
ling energy.  
  
Length:      7 feet  
Height:      8 feet  
Location:     Underground  
Dangerous?    yes  
Number of Hours to Care for Creature:      $ 0  
Cost Per Hour of Taking Care of Creature:  $ 1000  
Cost to Feed Creature:                     $ 700  
Grooming & Supplies Cost:                  $ 0  
  
-----  
CREATURE 2:  
Name:      Banshee  
Description:  
The English Banshee is a fairy woman who wails when death is approaching.They do  
not cause death, only mourn it. Banshees are almost always female, and are usu  
ally seen with long, dark, black hair and pale cheeks. Their eyes also are usual  
ly red from crying.  
  
Length:      0 feet  
Height:      0 feet  
Location:     Irish  
  
-----  
Dangerous?    no  
Number of Hours to Care for Creature:      $ 1  
Cost Per Hour of Taking Care of Creature:  $ 15.5  
Cost to Feed Creature:                     $ 0  
Grooming & Supplies Cost:                  $ 25  
  
-----  
CREATURE 3:  
Name:      Sasquatch  
Description:  
Sasquatch is large, hairy, man-shaped creature that live in the forests of North  
America, especially around the border of the US and Canada. Like their cousin  
in the Himalayas (known as Yeti) they have been glimpsed at by many, but no proo  
f of their existence has been found except the odd fur and footprints. Sasquatc  
h is also known as Bigfoot.  
  
Length:      0 feet  
Height:      7.5 feet  
Location:     US and Canada  
Dangerous?    yes  
Number of Hours to Care for Creature:      $ 15.75  
Cost Per Hour of Taking Care of Creature:  $ 8.5  
Cost to Feed Creature:                     $ 265.75  
Grooming & Supplies Cost:                  $ 12.4  
  
-----  
CREATURE 4:  
Name:      Troll
```

```

Description:
Ugly and big. Sometimes smell bad.

Length:      0 feet
Height:      8.2 feet
Location:     Wooded areas
Dangerous?   yes
Number of Hours to Care for Creature:      $ 8
Cost Per Hour of Taking Care of Creature:  $ 1
Cost to Feed Creature:                     $ 1
Grooming & Supplies Cost:                  $ 3.45

-----

CREATURE 5:
Name:      Unicorn
Description:
Big white horse with a horn. Can teleport and only allows the purist of people
to approach it. Deadly when angry. Can be calmed down with skittles.

Length:      16 feet
Height:      7 feet
Location:     England
Dangerous?   yes
Number of Hours to Care for Creature:      $ 15
Cost Per Hour of Taking Care of Creature:  $ 100
Cost to Feed Creature:                     $ 20
Grooming & Supplies Cost:                  $ 30.15

```

## EXTRA CREDIT OPTION

For 5 points extra credit, your program can do word wrapping for the description to make it much more readable. In other words, each word is not split up – they are kept together. An example is below. To get the full points, your description would have to keep the words together and each line should have an indentation like the example below.

```

-----
CREATURE 2:
Name:      Banshee
Description:
The English Banshee is a fairy woman who wails when death is
approaching. They do not cause death, only mourn it. Banshees
are almost always female, and are usually seen with long,
dark, black hair and pale cheeks. Their eyes also are usually
red from crying.

Length:      0 feet
Height:      0 feet
Location:     Irish
Dangerous?   no
Number of Hours to Care for Creature:      $ 1
Cost Per Hour of Taking Care of Creature:  $ 15.5
Cost to Feed Creature:                     $ 0
Grooming & Supplies Cost:                  $ 25

-----
CREATURE 3:

```

## PRINTSTATISTICS FUNCTION

The `printStatistics()` function is a void function and it contains two parameters: the current number of creatures in the Creatures array and the Creatures array.

This function should print out the total cost of each creature in a table. To figure out the cost of a creature, use the formula:

**`COST = numHours * costPerHour + foodCost + materialCost`**

Then, this function should also keep a running total of the total cost of all the creatures.

The table printed should look like the one below. Use stream manipulators to print out the table in nice columns!



## COST OF EACH CREATURE FOR ONE WEEK:

CREATURE		COST
Beholder	\$	700.00
Banshee	\$	40.50
Sasquatch	\$	412.02
Troll	\$	12.45
Unicorn	\$	1550.15
TOTAL COST:	\$	2715.12

### SAVECREATURESTOFILE FUNCTION

The `saveCreaturesToFile()` function is a void function and it contains two parameters: the current number of creatures in the Creatures array and the Creatures array.

The function should ask the user what the name of the file that they wish to save their creatures to. The function should then open that file and write out all the creature data in the following order:

- Name
- Description
- Avg. Length
- Avg. Height
- Location
- Dangerous
- numHours
- costPerHour
- foodCost
- MaterialCost

**There should be no newlines or endlines in the file. All data should be separated by a pound sign (#) instead of a space.**

After printing all data from the Creatures array to the file, this function should print out a message to standard output to tell Hagrid that all creatures were successfully saved. An example of this output and the resulting text file is below.

```
What is the name of the file you want to save your creatures to?  
FILENAME:  my_creatures.txt
```

```
Your creatures were successfully saved to the my_creatures.txt file.
```

```
GOODBYE!
```

```
my_creatures.txt - Notepad
File Edit Format View Help

Beholder#Giant center eye and twelve eye stalks above it. It is a flying eyeball. Mouth full of
razor sharp teeth. Eye stalks shoot various beams of magical death-dealing
energy.#7#8#Underground#1#0#1000#700#0#Banshee#The English Banshee is a fairy woman who wails when
death is approaching.They do not cause death, only mourn it. Banshees are almost always female,
and are usually seen with long, dark, black hair and pale cheeks. Their eyes also are usually red
from crying.#0#0#Irish#0#1#15.5#0#25#Sasquatch#Sasquatch is large, hairy, man-shaped creature that
live in the forests of North America, especially around the border of the US and Canada. Like
their cousin in the Himalayas (known as Yeti) they have been glimpsed at by many, but no proof of
their existence has been found except the odd fur and footprints. Sasquatch is also known as
Bigfoot.#0#7.5#US and Canada#1#15.75#8.5#265.75#12.4#Troll#Ugly and big. Sometimes smell
bad.#0#8.2#Wooded areas#1#8#1#1#3.45#Unicorn#Big white horse with a horn. Can teleport and only
allows the purist of people to approach it. Deadly when angry. Can be calmed down with
skittles.#16#7#England#1#15#100#20#30.15#
```



#### EXTRA CREDIT – DELETE CREATURE FUNCTION

The `deleteCreature()` function has two parameters: the current number of creatures in the Creatures array and the Creatures array. This function returns the new number of creatures.

First, this function will say “The following is a list of all the creatures you take care of: ” and then it will say the name of each creature.

```
The following is a list of all the creatures you take care of:
Beholder
Banshee
Sasquatch
Troll
```

Then, your program will ask the user which one they want to remove.

```
What creature do you wish to remove?  
CREATURE NAME:
```

Your program should then read in the name and place it in a variable.

Then, find the subscript number of the creature that needs to be removed.

If the subscript is found, then you know there is a creature in the **Creatures** array by that name and that your program will be able to remove it. If your program cannot find the creature in the array, then you will tell the user that the creature could not be found.

```
What creature do you wish to remove?  
CREATURE NAME: Bob
```

```
Sorry, a creature by the name Bob could not be found.
```

Otherwise, this function should now overwrite the element with the creature to delete (x) with the next element in the array (x+1), moving each element after the deleted element to the left. Then decrement the number of creatures and inform the user.

```
What creature do you wish to remove?  
CREATURE NAME: Sasquatch
```

```
You have removed Sasquatch.
```



## HOW DO I COMPILE A PROGRAM WITH MULTIPLE FILES?

Compiling a C++ program to create an executable file is a two-step process. You first compile each source file (including any header files) into an object file. Then, you link all the object files together into an executable file.

Here are the steps to do that for this program below. The '-I' stands for include and so the '-I ./' means that we are including all header files (.h files) in our current directory in this compile.

This line will compile the Creatures.cpp file, including all the .h files in the current directory (Creatures.h), and then create the object file (Creatures.o)

```
g++ -I ./ -c Creatures.cpp
```

This line will compile the Functions.cpp file, including all the .h files in the current directory (Creatures.h), and then create the object file (Functions.o)

```
g++ -I ./ -c Functions.cpp
```

This line will link the object files specified (Creatures.o & Functions.o) into the output file (executable file) named Program3.cpp.

```
g++ Creatures.o Functions.o -o Program3
```

The "shortcut" way to do all of this in one line instead of the three lines above is demonstrated below. This line will compile the listed source files including all the .h files in the current directory (Creatures.h) and then link the resulting object files into the output file (executable file) named Program3.cpp

```
g++ -I ./ Creatures.cpp Functions.cpp -o Program3
```

To run your program with the test case provided, you should do the following:

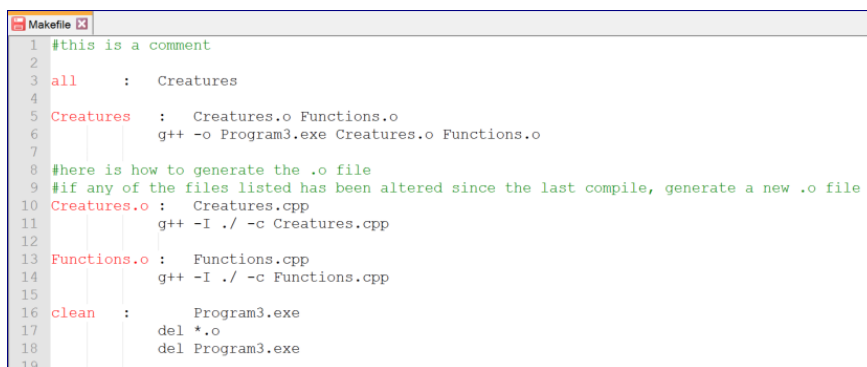
```
Program3 < TEST_CASE1.txt
```

## WHAT IS A MAKEFILE?

If you have a Mac then I highly recommend trying out the given Makefile on the lab desktop computers during your lab one day.

The Windows compiler TDM-GCC comes with the make utility, which allows you to run a makefile to compile, recompile, and link a program whenever changes are made to the source or header files. The more files you have in your program, the better it is to create a makefile. All you have to do is open notepad or other simple text editor and create your makefile. For information on how to make your own Makefile, read under "ADDITIONAL HELPFUL MATERIAL" in your zyBooks in section 11.1 called Makefiles. Name the file makefile and do not give it an extension. Then, you should be able just to type make in the command prompt and the lines in your makefile will execute.

Here is the makefile I provided for you:



```
1 #this is a comment
2
3 all : Creatures
4
5 Creatures : Creatures.o Functions.o
6           g++ -o Program3.exe Creatures.o Functions.o
7
8 #here is how to generate the .o file
9 #if any of the files listed has been altered since the last compile, generate a new .o file
10 Creatures.o : Creatures.cpp
11             g++ -I ./ -c Creatures.cpp
12
13 Functions.o : Functions.cpp
14             g++ -I ./ -c Functions.cpp
15
16 clean :
17        del Program3.exe
18        del *.o
19        del Program3.exe
```

Here is an example of running my makefile from the command prompt:

```
C:\Windows\System32\cmd.exe
C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE - Magical Cr
eature Zoo program - due November 13\solution with extra credit>make
g++ -I ./ -c Creatures.cpp
g++ -I ./ -c Functions.cpp
g++ -o Program3.exe Creatures.o Functions.o

C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE - Magical Cr
eature Zoo program - due November 13\solution with extra credit>
```

This produces my executable file named Program3.exe.

```
C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE - Magical Cr
eature Zoo program - due November 13\solution with extra credit>dir
Volume in drive C is windows
Volume Serial Number is E0FE-3288

Directory of C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE
E - Magical Creature Zoo program - due November 13\solution with extra credit
10/18/2017 04:22 PM <DIR> .
10/18/2017 04:22 PM <DIR> ..
10/18/2017 03:46 PM 117 build.bat
10/18/2017 12:46 PM 1,381 Creatures.cpp
10/18/2017 12:28 PM 1,267 Creatures.h
10/18/2017 04:22 PM 6,350 Creatures.o
10/18/2017 03:11 PM 12,915 Functions.cpp
10/18/2017 04:22 PM 30,839 Functions.o
10/18/2017 04:17 PM 443 Makefile
04/12/2010 08:18 AM 981 my_creatures.txt
10/18/2017 04:22 PM 2,858,423 Program3.exe
9 File(s) 2,912,716 bytes
2 Dir(s) 310,906,306,560 bytes free

C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE - Magical Cr
```

I can then use the clean command to remove the .o and .exe files if I want to:

```
C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE - Magical Cr
eature Zoo program - due November 13\solution with extra credit>make clean
del *.o
del Program3.exe

C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE - Magical Cr
eature Zoo program - due November 13\solution with extra credit>dir
Volume in drive C is windows
Volume Serial Number is E0FE-3288

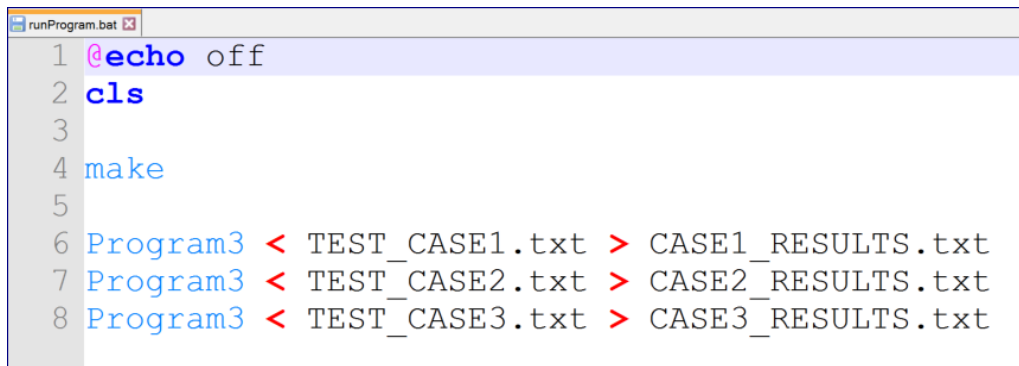
Directory of C:\Users\acrockett\Desktop\CSC1300 Fall 2017\PROGRAMS\PROGRAM THREE
E - Magical Creature Zoo program - due November 13\solution with extra credit
10/18/2017 04:24 PM <DIR> .
10/18/2017 04:24 PM <DIR> ..
10/18/2017 03:46 PM 117 build.bat
10/18/2017 12:46 PM 1,381 Creatures.cpp
10/18/2017 12:28 PM 1,267 Creatures.h
10/18/2017 03:11 PM 12,915 Functions.cpp
10/18/2017 04:17 PM 443 Makefile
04/12/2010 08:18 AM 981 my_creatures.txt
6 File(s) 17,104 bytes
2 Dir(s) 310,908,878,848 bytes free
```

*Notice they are gone*

## WHAT IS A BATCH FILE?

If you have a Mac then I highly recommend trying out the given batch file on the lab desktop computers during your lab one day.

A Windows batch file is a script file containing a list of instructions to be carried out in turn by the command prompt. The batch file I provided for you is called runProgram.bat. Here are it's contents:

A screenshot of a Windows batch file editor window titled 'runProgram.bat'. The window contains eight lines of code. Line 1: '@echo off' (the '@' is grey, 'echo' is blue, 'off' is black). Line 2: 'cls' (blue). Line 3: (empty). Line 4: 'make' (blue). Line 5: (empty). Line 6: 'Program3 < TEST\_CASE1.txt > CASE1\_RESULTS.txt' (Program3 is blue, < is red, TEST\_CASE1.txt is black, > is red, CASE1\_RESULTS.txt is black). Line 7: 'Program3 < TEST\_CASE2.txt > CASE2\_RESULTS.txt' (Program3 is blue, < is red, TEST\_CASE2.txt is black, > is red, CASE2\_RESULTS.txt is black). Line 8: 'Program3 < TEST\_CASE3.txt > CASE3\_RESULTS.txt' (Program3 is blue, < is red, TEST\_CASE3.txt is black, > is red, CASE3\_RESULTS.txt is black).

```
1 @echo off
2 cls
3
4 make
5
6 Program3 < TEST_CASE1.txt > CASE1_RESULTS.txt
7 Program3 < TEST_CASE2.txt > CASE2_RESULTS.txt
8 Program3 < TEST_CASE3.txt > CASE3_RESULTS.txt
```

Basically you should be able to double click on the batch file and it should compile & run your program 3 the test cases as input and then printing out to the results text files instead of standard output.