# CITP 155 – Programming I

## Switches Lab

## Graded Activity (15 points)

Provide any screen shots using the Snipping Tool and selecting only the relevant portion of the screen (instead of the entire screen). Provide any answers using a blue font. Please note that part of being a good programmer is being precise. If you have typos in your code, such as the words that are supposed to be displayed, you will not receive full credit.

KEEP ALL OF YOUR SOLUTION AND PROJECT FILES THROUGHOUT THE DURATION OF THE CLASS!

Concepts taken from <https://csharp.net-tutorials.com/control-structures/switch-statement/>

**Questions (2 points)**

1. Describe a switch statement.

A switch statement is similar to an IF statement, but is more specific to the case provided. So it will look to see if the input given matches an expected input (usually a char or int) and if it does will run a bit of code.

1. In a switch statement, what statement or command is placed at the end of each case?

break; is usually placed at the end of a statement so the switch statement will end. If break; is left out then you will have multiple cases running their code which can be

1. In a switch statement, what statement or command is used in case no other scenarios have been met?

default: is used when no other case has been met. This is similar to an ELSE statement.

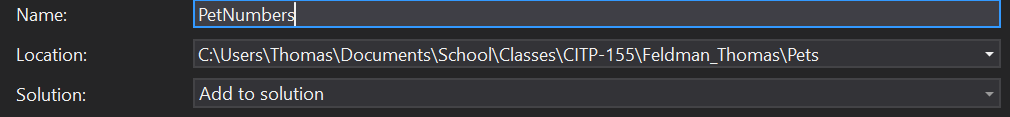
**Create a New Project (1 points)**

Using Visual Studio, open you **Pets** solution.

Create a new Visual C# Console Application project which will be part of the existing **Pets** solution.

Project Name: PetNumbers

Provide a screen shot of the New Project screen before clicking OK.



**Open and Modify Your C# Program**

Using the coding examples found at <https://csharp.net-tutorials.com/control-structures/switch-statement/> as a guide, create a program that does the following:

* Declare a variable that is an **int** data type. Name the variable **petNo**. Initially set the variable to **0**.
* Using the console, ask the end user this question:  
  **How many pets do you have?**
* Store the response from the end user in the petNo variable. Be sure to check to make sure it is a number as you are capturing the input.
* Use a switch statement to evaluate the number entered.
  + If the number is 0, use the console to display:  
    **Maybe you will get some pets someday soon!**
  + If the number is 1, use the console to display:  
    **Say hello to your special friend from xx!**where xx is your first name.
  + If the number is 2, use the console to display:  
    **Two is a nice number of pets!**
  + If the number is 3, use the console to display:  
    **You must really like pets!**
  + If the number is 4, use the console to display:  
    **Four is a lot of pets!**
  + For the default option for your switch statement, you are going to do two things, 1) check to see if the number is less than zero and 2) suppose the number is nothing else that has been checked for (so it’s greater than 4).  
    To do this, you will need to use the following code:

if (petNo < 0)

Console.WriteLine("That's not a valid number of pets!");

else

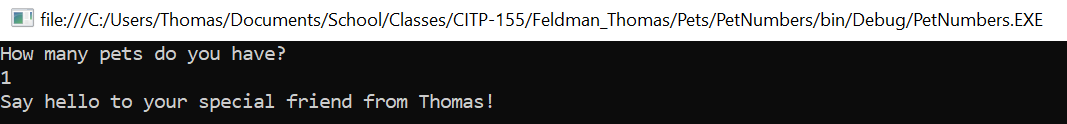
Console.WriteLine("You have so many pets!");

break;

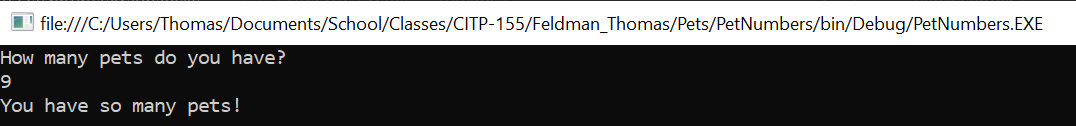
* Make sure the console pauses at the end so the messages can be read by the end user.
* Test your program. Be sure to test with valid and invalid input (negative numbers, letters, 0, 1, 2, 3, 4, and higher numbers).

**Provide Results (12 points)**

Provide a screen shot of your testing when you entered 1 as the response. The screen shot should include the title bar of the console with the full path of the program. It should also include three lines of white text.



Provide a screen shot of your testing when you entered 9 as the response. The screen shot should include the title bar of the console with the full path of the program. It should also include three lines of white text.



Copy and paste the lines of code from your program here. This is not a screen shot. This is code I can copy and paste to run on my own. This should be 35-45 lines of code, depending on how many blank lines you left in between other lines of code.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace PetNumbers

{

class Program

{

static void Main(string[] args)

{

int petNo = 0;

Console.WriteLine("How many pets do you have?");

petNo = Convert.ToInt32(Console.ReadLine());

switch (petNo)

{

case 0:

Console.WriteLine("Maybe you will get some pets someday soon!");

break;

case 1:

Console.WriteLine("Say hello to your special friend from Thomas!");

break;

case 2:

Console.WriteLine("Two is a nice number of pets!");

break;

case 3:

Console.WriteLine("You must really like pets!");

break;

case 4:

Console.WriteLine("Four is a lot of pets!");

break;

default:

if (petNo < 0){

Console.WriteLine("That's not a valid number of pets!");

}

else {

Console.WriteLine("You have so many pets!");

}

break;

}

Console.ReadLine();

}

}

}