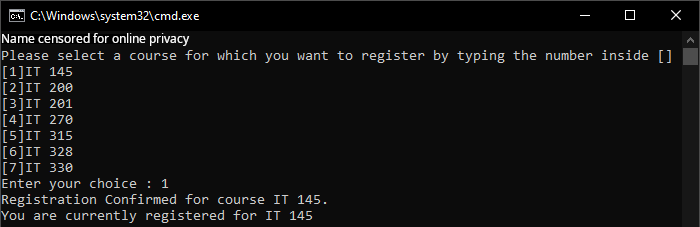
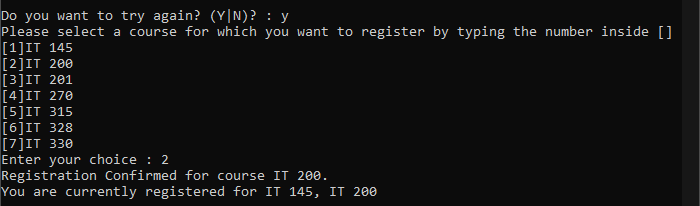
**Screenshots (was all originally just one big screenshot, but I split it for readability’s sake):**



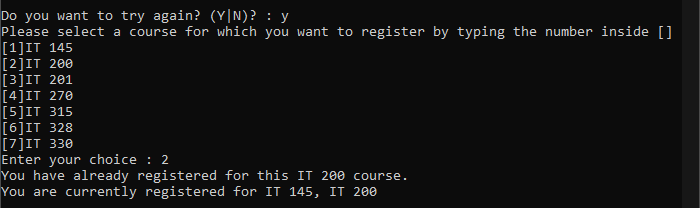
The above screenshot proves first class registration functionality.

---------------------------------------------------------------------------------------------------------------------

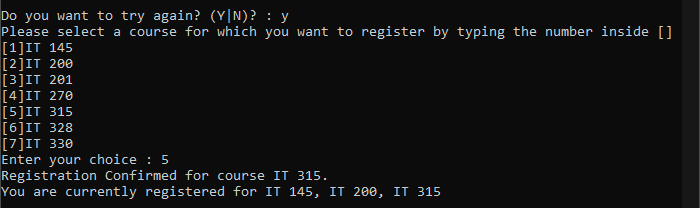


The above screenshot proves second class registration functionality.

---------------------------------------------------------------------------------------------------------------------

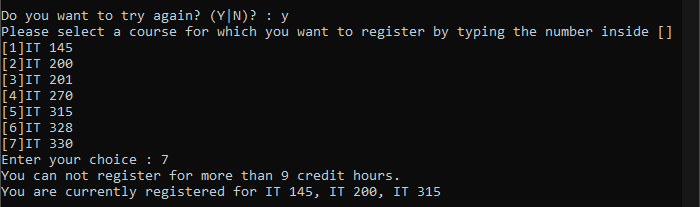


The above screenshot proves redundancy checking functionality.



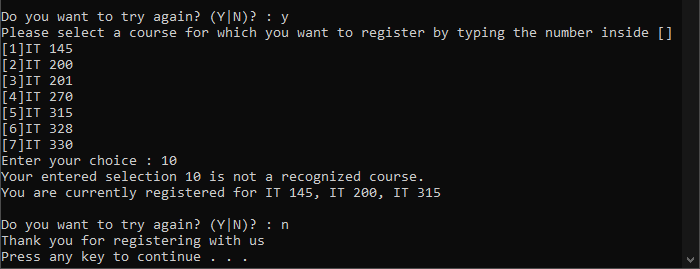
The above screenshot proves third class registration functionality.

---------------------------------------------------------------------------------------------------------------------



The above screenshot proves overdraw checking functionality.

---------------------------------------------------------------------------------------------------------------------



The above screenshot proves checking for invalid numbers and exiting the loop functionality.

**Code:**  
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleRegisterStudent

{

class Program

{

// Dictionary to retrieve class info from.

Dictionary<int, string> classListings = new Dictionary<int, string>()

{

// Please do not assign `0` as a key. That will break much of the logic.

{ 1, "IT 145" },

{ 2, "IT 200" },

{ 3, "IT 201" },

{ 4, "IT 270" },

{ 5, "IT 315" },

{ 6, "IT 328" },

{ 7, "IT 330" }

};

// Implemented enums for code readability.

enum Validation

{

Unrecognized,

Redundant,

Overdrawn,

Valid

}

static void Main(string[] args)

{

(new Program()).Run();

}

void Run()

{

int choice;

int firstChoice = 0, secondChoice = 0, thirdChoice = 0;

int totalCredit = 0;

string yesOrNo = "";

System.Console.WriteLine("BunkWire2X8");

do

{

WritePrompt();

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

// Changed to utilize enums.

switch (ValidateChoice(choice, firstChoice, secondChoice, thirdChoice, totalCredit))

{

case Validation.Unrecognized:

Console.WriteLine("Your entered selection {0} is not a recognized course.", choice);

break;

case Validation.Redundant:

// Fixed spelling error.

Console.WriteLine("You have already registered for this {0} course.", ChoiceToCourse(choice));

break;

case Validation.Overdrawn:

Console.WriteLine("You can not register for more than 9 credit hours.");

break;

case Validation.Valid:

Console.WriteLine("Registration Confirmed for course {0}.", ChoiceToCourse(choice));

totalCredit += 3;

if (firstChoice == 0)

firstChoice = choice;

else if (secondChoice == 0)

secondChoice = choice;

else if (thirdChoice == 0)

thirdChoice = choice;

break;

}

WriteCurrentRegistration(firstChoice, secondChoice, thirdChoice);

Console.Write("\nDo you want to try again? (Y|N)? : ");

yesOrNo = (Console.ReadLine()).ToUpper();

} while (yesOrNo == "Y");

Console.WriteLine("Thank you for registering with us");

}

private void WritePrompt()

{

Console.WriteLine("Please select a course for which you want to register by typing the number inside []");

// Print statement edited to retrieve directly from classListings dictionary.

foreach (int num in classListings.Keys)

Console.WriteLine($"[{num}]{classListings[num]}");

Console.Write("Enter your choice : ");

}

// Changed to utilize enums.

Validation ValidateChoice(int choice, int firstChoice, int secondChoice, int thirdChoice, int totalCredit)

{

// Rewrote validation to utilize dictionary.

if (!classListings.ContainsKey(choice))

return Validation.Unrecognized;

// Changed 'and' bool modifiers to 'or' bool modifiers.

else if (choice == firstChoice || choice == secondChoice || choice == thirdChoice)

return Validation.Redundant;

// Made sure to account for the future credit, not current credit.

// Technically, I could write this as `totalCredit > 6`, but this is more readable.

else if (totalCredit + 3 > 9)

return Validation.Overdrawn;

return Validation.Valid;

}

// Rewrote code to be slightly less redundant and use finer logic.

void WriteCurrentRegistration(int firstChoice, int secondChoice, int thirdChoice)

{

if (firstChoice == 0)

return;

string classes = $"{ChoiceToCourse(firstChoice)}";

if (secondChoice != 0)

{

classes += $", {ChoiceToCourse(secondChoice)}";

if (thirdChoice != 0)

classes += $", {ChoiceToCourse(thirdChoice)}";

}

Console.WriteLine($"You are currently registered for {classes}");

}

// Simplified function to use the new classListings dictionary.

string ChoiceToCourse(int choice)

{

if (classListings.ContainsKey(choice))

return classListings[choice];

return "";

}

}

}