

CSci 3081W: Program Design and Development
Homework 1 – C++ Basics
Fall 2023

This document is subject to minor changes (grammar, spelling, and clarity) until Friday, September 15th at 3:20 pm.

Due Date:

- **Full Assignment** - Sunday, October 1st, 2023, before 11:59 pm.

Overview: This homework will be a set of problems that you answer with code in C++. The concepts in every problem are covered in the Zybooks assignments.

Instructions: This homework is a coding assignment. To receive credit for this homework, you will need to submit a set of .cc/.h files to Gradescope. **Please do not zip them! This is an individual assignment. Feel free to discuss high level concepts with your classmates, but code sharing of any kind is not allowed and will result in a zero on the assignment.** All code must be able to run properly on a CSE labs machine (use [vole](#), [ssh](#), or in person to validate).

Problem 1: FakeGPT (10 points)

In this problem, you will write C++ programs that pretend (badly) to be ChatGPT. Your code will take in text and decide whether or not the text contains a question. If the text does contain a question, your code will tell the user that they should Google that question. Otherwise, your code will tell the user that since their input is not a question, it doesn't know how to respond. Additionally, since AI systems are growing more powerful, we want to teach the user to be more polite to them. Some sample inputs and outputs are provided to you below. ****NOTE: your code should output the exact phrasing for each case in parts B and C.****

Example inputs/outputs:

Input:

WHAT day is today?

Output:

I'm not sure, you should google that! Your questions should also be more polite.

Input:

please tell me what day is today?

Output:

I'm not sure, you should google that! Thanks for asking so politely.

Input:

Please answer this question for me

Output:

That is not a question! But you said it so nicely.

Input:

Answer this question for me

Output:

That is not a question! And you should be more polite.

- In a file named `fakeGPT_a.cc`, write a C++ program that reads in a string input of variable length and separates it into individual words (strings) stored in a Vector. Your code will output the length of that Vector. *****TIP: You can (and should) copy/paste this code to use for part B and C to make your life easier!*****
- In a different file named `fakeGPT_b.cc`, write a C++ program that reads in a string input of variable length and uses if/else branches to decide if the input is a question and if the question is polite. The program should then print out appropriate responses. *****NOTE: this file should run independently from `fakeGPT_a.cc` and not depend on it.*****

The string input will be a question if it contains any of the following words:

who, what, where, when, why, or how

The string input is polite if it contains any of the following words:

please or may

- In another separate file named `fakeGPT_c.cc`, replicate the functionality of `fakeGPT_b.cpp` but this time using a switch statement. *****NOTE: this file should run independently from `fakeGPT_a.cc` and `fakeGPT_b.cc` and should not depend on either.*****

Submission: Submit three files to Gradescope for this problem: `fakeGPT_a.cc`, `fakeGPT_b.cc`, and `fakeGPT_c.cc`

Problem 2: IP Addresses (20 points)

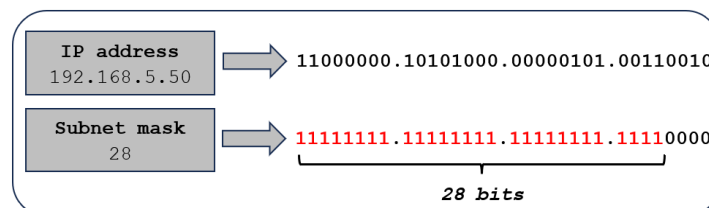
You are tasked with designing a program that will calculate the network address, broadcast address, and valid IP address range for a given IPv4 address and subnet mask.

Overview:

If given an IP address and subnet mask, we can calculate the network address and broadcast address as follows:

Step 1: Convert the addresses into binary format

[Note: For subnet mask, the user will enter a value between 1-32]



Step 2: Use AND operation between IP address (binary) and subnet mask (binary) to calculate the Network address

IP address (Decimal Notation)	192	168	5	50
IP address (Binary Equivalent)	11000000	10101000	00000101	00110010
Subnet Mask	11111111	11111111	11111111	11110000
AND	11000000	10101000	00000101	00110000
Network Address	192	168	5	48

Step 3: Use OR operation between IP address (binary) and subnet mask (inverse binary) to calculate the Broadcast address

IP address (Decimal Notation)	192	168	5	50
IP address (Binary Equivalent)	11000000	10101000	00000101	00110010
Subnet Mask (Inverse)	00000000	00000000	00000000	00001111
OR	11000000	10101000	00000101	00111111
Broadcast Address	192	168	5	63

Step 4: Valid IP address range is determined by the network address (upper bound) and broadcast address (lower bound)

Instructions:

- In a file named `IPAddress.cc`, create a class named `IPAddress` with the following attributes and methods:
 - A vector `ipOctets` that stores the four octets of the IP address
 - A vector `smOctets` that stores the four octets of the subnet mask
 - A constructor that initializes the IP address and subnet mask octets
 - A function `calculateNetworkAddress` that calculates the network address as vector output
 - A function `calculateBroadcastAddress` that calculates the broadcast address as vector output [Can be combined with previous function]
 - A function `calculateValidIPAddressRange` that calculates the valid IP addresses as a pair of vector integers

IPAddress
- ipOctets: std::vector<int> - smOctets: std::vector<int>
+ IPAddress(std::vector<int>, std::vector<int>) + calculateNetworkAddress(): std::vector<int> + calculateBroadcastAddress(): std::vector<int> + calculateValidIPAddressRange(): std::vector<int>

2. In a main function in the same file, prompt the user to input an IPv4 address and a subnet mask. Create an `IPAddress` object with the provided IP address and a subnet mask. Use the methods of the `IPAddress` class to calculate and display the network address, broadcast address and valid IP address range.
3. Ensure the program handles input validation, such as checking that the provided IP address and subnet mask are in valid formats (e.g., "xxx.xxx.xxx.xxx")

Examples:

Input:

```
Enter an IPv4 address [xxx.xxx.xxx.xxx] (or Type q to exit):
192.168.5.50
Enter a subnet mask [1-32] (or Type q to exit):
28
```

Output:

```
Network Address: 192.168.5.48
Broadcast Address: 192.168.5.63
Valid IP Address Range: 192.168.5.48 - 192.168.5.63
```

Submission: Submit one file to Gradescope for this problem: `IPAddress.cc`

Problem 3: Planes (70 points)

In this problem, you will be documenting, implementing existing functionality, and adding new functionality to the Airplanes & Airports codebase. You will be using this GitHub repository for this portion of the homework: <https://github.com/umn-csci-3081-f23/public-hw01>. Please read all files carefully. There are three separate parts to complete in this portion of the assignment. All the instructions for this problem are in [main.cc](#).

Submission: Submit your four modified files to Gradescope: `Airplane.cc`, `Airplane.h`, `AirportAndGate.h`, and `main.cc`. **Please do not zip them!**