

# Introduction to C++ Assignment

## Assignment A01

Welcome to your first C++ assignment. This is a “no ai” assignment, meaning that your source code should be a genuine reflection of your personal effort and understanding. None of the code you submit should be generated by an ai tool like ChatGPT, Github Copilot, or other similar tools. The goal here is to ensure that your C++ foundations are solid as a rock.

Please read “Part 4” before starting the assignment, as it asks you to keep track of how long each program took to write. It is also important that you read through the rubric before you begin. Specifically, **there are new “code cleanliness” and “code commenting” requirements.**

Before you start the assignment be sure to read through the following sections of our notes:

- [Console I/O](#)
- [Basic Math](#)
- [Booleans and Decisions](#)
- [Loops](#) (Note that this section goes further into I/O, specifically validating input.)
- [Functions](#)

Alternatively, you could watch [the recorded lecture on these topics from last year](#).

**Final thing:** Please don’t make any recursive calls in your programs. If you’re tempted to use recursion (i.e. to have a function call itself) use a loop instead.

## Part 1 - Kitty Kata

Create a new C++ project with a single main.cpp file that prints every number from 1 to 100 to separate lines in the console using cout statements. Your code can use either a while loop or a for loop to accomplish this. Next, modify your code so that for [multiples](#) of three you print “Kitty” instead of the number and for the multiples of five you print “Kata”. For numbers which are multiples of both three and five you print “KittyKata”.

You’ve written this program before in Javascript. Feel free to look up the program you wrote and “port” it to C++.

*Sample Output:*

```
1
2
Kitty
4
```

```
Kata
Kitty
7
8
Kitty
Kata
11
Kitty
13
14
KittyKata
16
... etc up to 100
```

## Part 2 - CHIRP CHIRP CHIRP

The common field cricket chirps in direct proportion to the temperature.

Temperature in Fahrenheit =  $50 + (\text{chirpsPerMin} - 40) / 4$

Write a C++ program that accepts as input the number of cricket chirps in *fifteen seconds*, then outputs the current temperature in Fahrenheit.

Once you've got this working update the program to support outputting the temperature in both Fahrenheit and Celsius, given that  $^{\circ}\text{C} = (^{\circ}\text{F} - 32) * 5/9$

Lastly, note that crickets stop singing when the temperature is below 50 °F. Add a feature to your program warning users of invalid input if the output temperature is below this threshold.

You can [use this online calculator](#) to check your work. (*Careful, it defaults to outputting Celsius.*)

**Medium Hard Mode:** Refactor your code to make use of functions for key bits of program functionality. See [our function notes](#) and [this section of the recorded lecture from 2024](#).

**Hard Mode:** Refactor your code to ensure that all functions (other than your main() entry point) live in a separate \*.cpp file with associated forward declares in a \*.hpp header files. See [our notes on header files and functions](#) and [this section of the recorded lecture from 2024](#).

## Part 3 - Please Choose a Password

Write a program that:

- Prompts the user to choose a password that meets certain criteria (listed later).
- If the user's password does not meet the set criteria tell them what needs to change, and re-prompt them to choose a password.
- Do not overwhelm the user. Show them only one criteria issue at a time.

- Once the user's password meets the set criteria, ask them to type the password again.
- If their second confirmation password doesn't match their original password, start at the beginning again by re-prompting them for a new password.
- Once the user confirms their selected password, thank them and print out the password.

Password Criteria:

- The password must be at least 8 characters in length.
- The password must contain at least one lower case letter.
- The password must contain at least one upper case letter.
- The password must contain at least one number.

**Hard mode:** Research the `_getch` function found in `conio.h` and use it to hide the user's input. Asterisks should be printed out in place of the actual character that the user types.

**Extra hard mode:** Ensure that your program implements `hardmode`, but also supports using backspace. For example, a user should be able to type "aaaaAb1c98" and then backspace two times. After that only eight asterisks should be displayed on screen and the password would be "aaaaAb1c". When the user hits a backspace you'll need to detect that, and then remove a char from the end of the string. Separately, you'll also need to erase one of the asterisks by using the backspace escape character in a clever fashion.

## Part 4 – Reflection

Write a short reflection on this assignment. One or two paragraphs will be sufficient. Please include the following:

- What problems you encountered and/or discoveries you made while working on this assignment.
- Personal difficulty rating for the various problems: Easy, Medium, Hard
- Approximate length of time you worked on each problem. (This is not for me to judge you. Coding speed doesn't really matter. I just want to ensure that my assignments aren't too long on average.)

## Marking Rubric (19 marks)

### Part 1 – Kitty Kata (2 marks)

- Correctly uses a loop to print numbers, with clear logic and no errors.
- Accurately prints "Kitty," "Kata," and "KittyKata" for correct multiples.

### Part 2 – Chirp Chirp Chirp (5 marks)

- Accurately calculates temperature in Fahrenheit.
- Accurately calculates temperature in Celsius.

- Includes validation warning for temperatures below 50 Fahrenheit.
- You’ve implemented the “medium hard mode” functionality.
- You’ve implemented the “hard mode” functionality.

### Part 3 – Please Choose a Password (8 marks)

- Each of the four password criteria tests are successfully implemented. (4 marks)
- Only one criteria issue is displayed at a time. (For example, if there are two criteria issues, don’t show the second until the user has resolved the first.)
- The password retype confirmation logic is correctly implemented and the selected password is output to the console.
- You’ve implemented the “hard mode” functionality.
- You’ve implemented the “extra hard mode” functionality.

### Code Cleanliness (Marks deducted for infractions.)

1. **Consistent Formatting:** The code follows consistent indentation and spacing rules, making it easy to read.
2. **No Recursive Calls:** None of your program include functions that call themselves, or collections of functions that call themselves recursively.
3. **Descriptive Naming:** Variables, functions, and classes are named descriptively, reflecting their purpose or use.

### Code Commenting (Marks deducted for infractions.)

1. **Function Documentation:** Each function includes a comment describing its purpose, input parameters, return values, and any side effects.
2. **Purposeful Comments (When Code Isn’t Obvious):** Comments explain the why, not just the what. They describe the intent or reasoning behind complex or non-obvious code.
3. **Avoids Over-Commenting:** The code isn’t cluttered with obvious comments that state the obvious (e.g., `int x = 10; // Assign 10 to x`).

### Reflection and Overall Effort and Creativity (4 marks)

Overall	Excellent (2)	Acceptable (1)	Poor (0)
Reflection	Reflection is insightful, clearly written, and addresses all required points.	Reflection is complete but may lack depth or miss some of the required points.	Reflection is missing or fails to address many of the required points.

<b>Overall Effort and Creativity</b>	Student went above and beyond, implementing additional features or showing creativity.	Meets basic expectations with a reasonable level of effort.	Minimal effort, basic implementation with little creativity.
--------------------------------------	--	---	--

**Note to Self (Where Self == Kyle):** Integrate the hard mode and extra hard mode features into the assignment proper. Think harder about overall effort and creativity marks.