- Solution

In this lesson, we'll discuss the solution to the problem discussed in the previous lesson.

WE'LL COVER THE FOLLOWING

Solution

```
#include <iostream>
#include <string>

int main(){

  std::cout << std::endl;

  std::string str;

  // std::string::size_type idx= str.find("no");
  auto idx = str.find("no");

  if (idx == std::string::npos) std::cout << "no not found " << std::endl;

  std::cout << std::endl;

  str = {"dkeu84kf8k48kdj39kdj74945du942"};
  std::string str2{"84"};

  std::cout << "str: " << str <<  std::endl;
  std::cout << "str2: " << str2 << std::endl;

  std::cout << "str.find('8'): " << str.find('8') << std::endl;
  std::cout << "str.rfind('8'): " << str.rfind('8') << std::endl;
  std::cout << "str.find('8', 10): " << str.find('8', 10) << std::endl;
  std::cout << "str.find(str2): " << str.find(str2) << std::endl;
  std::cout << "str.rfind(str2): " << str.rfind(str2) << std::endl;
  std::cout << "str.find(str2, 10): " << str.find(str2, 10) << std::endl;

  std::cout << std::endl;

  str2 = "0123456789";
  std::cout << "str: " << str << std::endl;
  std::cout << "str2: " << str2 << std::endl;
```

```
  std::cout << "str.find_first_of(678): " << str.find_first_of("678") << std::endl;
  std::cout << "str.find_last_of(678): " << str.find_last_of("678") << std::endl;
  std::cout << "str.find_first_of(678, 10): " << str.find_first_of("678", 10) << std::endl;

  std::cout << "str.find_first_of(str2): " << str.find_first_of(str2) << std::endl;
  std::cout << "str.find_last_of(str2): " << str.find_last_of(str2) << std::endl;
  std::cout << "str.find_first_of(str2, 10): " << str.find_first_of(str2, 10) << std::endl;

  std::cout << std::endl;

  std::cout << "str: " << str << std::endl;
  std::cout << "str2: " << str2 << std::endl;

  std::cout << "str.find_first_not_of(678): " << str.find_first_not_of("678") << std::endl;
  std::cout << "str.find_last_not_of(678): " << str.find_last_not_of("678") << std::endl;
  std::cout << "str.find_first_not_of(678, 10): " << str.find_first_not_of("678", 10) << std:
  std::cout << "str.find_first_not_of(str2): " << str.find_first_not_of(str2) << std::endl;
  std::cout << "str.find_last_not_of(str2): " << str.find_last_not_of(str2) << std::endl;
  std::cout << "str.find_first_not_of(str2, 10): " << str.find_first_not_of(str2, 10) << std:

  std::cout << std::endl;

}
```

Explanation

In the above code, we have used all variations of the `find` function. We have taken two strings, i.e., `str` and `str2` for checking all the variations.

In the next lesson, we'll discuss some tools that can be used to modify and manipulate strings.