1 Variadic Templates

Since C++11, templates can accept a variable number of template parameters. This feature, known as **variadic templates**, allows you to pass an arbitrary number of arguments of arbitrary types to a template. It is especially useful in scenarios where you need flexibility in the number and types of arguments. For example, you can use variadic templates to create a function that prints an arbitrary set of objects:

```
#include <iostream>

void print() {}

template<class T, class... Types>
void print(T first, Types... args)
{
    std::cout << first << std::endl;
    print(args...);
}

int main()
{
    print(5, 6, 7.0, "Hello, World!");
}
```

If one or more arguments are passed to `print()`, the function template is used. `print()` calls itself recursively for the remaining arguments. The remianing arguments names `args` are called a **parameter pack**.