Programming Languages and Tools: Programming with C++ CS:3210:0003

Lecture/Lab #3

WE WANT YOUR FEEDBACK

HELP IMPROVE COMPUTER SCIENCE AT THE UNIVERSITY OF IOWA!

TAKE THE ANNUAL COMPUTING RESEARCH ASSOCIATION (CRA) SURVEY AND GET ENTERED TO **WIN A $20 AMAZON GIFT CARD (10 WINNERS)**

SCAN THE QR CODE TO TAKE THE SURVEY

WiCS @UIOWA
Women in Computing Sciences

IOWA
Department of Computer Science

Building a community of technical women at the University of Iowa!

- Informative tech talks
- Career-building workshops
  - Resume, interview, career fair, etc.
- Fun social events
  - Game Room Night, Cookie Decorating, Pottery etc.

Intro Meeting: Wednesday 1/24, 6:30PM - 7:30PM, MacLean Hall Room 113

@uiowawics

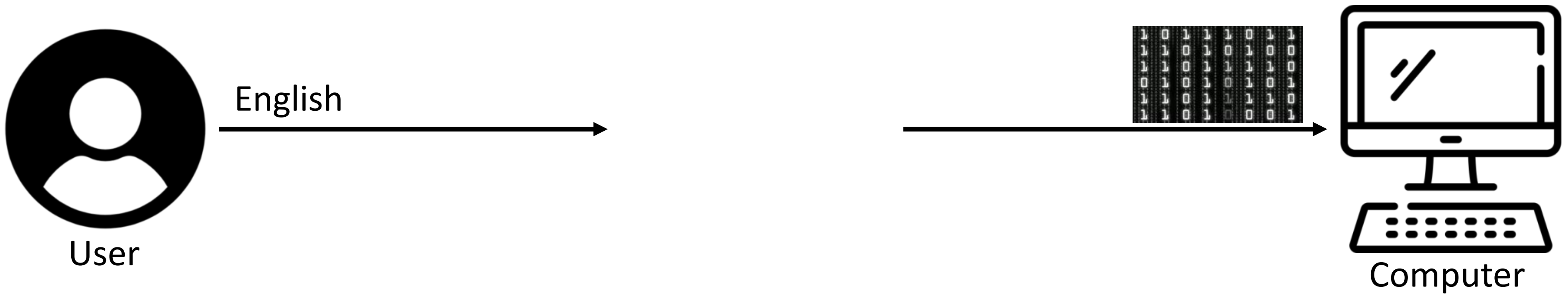
Individuals with disabilities are encouraged to attend all University of Iowa-sponsored events. If you are a person with a disability who requires a reasonable accommodation in order to participate in this program, please contact Annalisa Karacay in advance at annalisa-karacay@uiowa.edu

Computer Interface

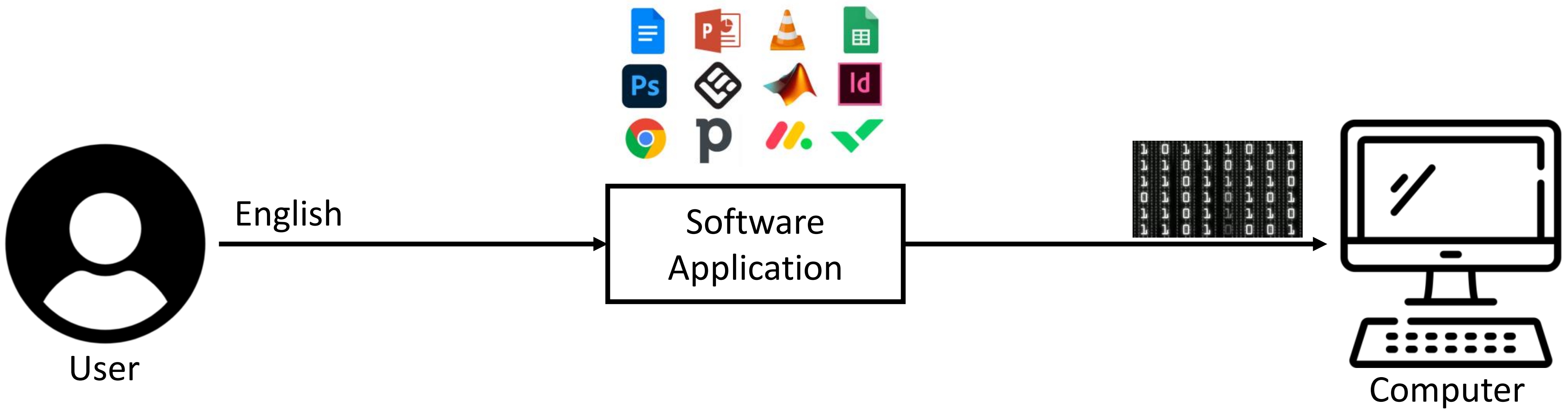
User

Computer

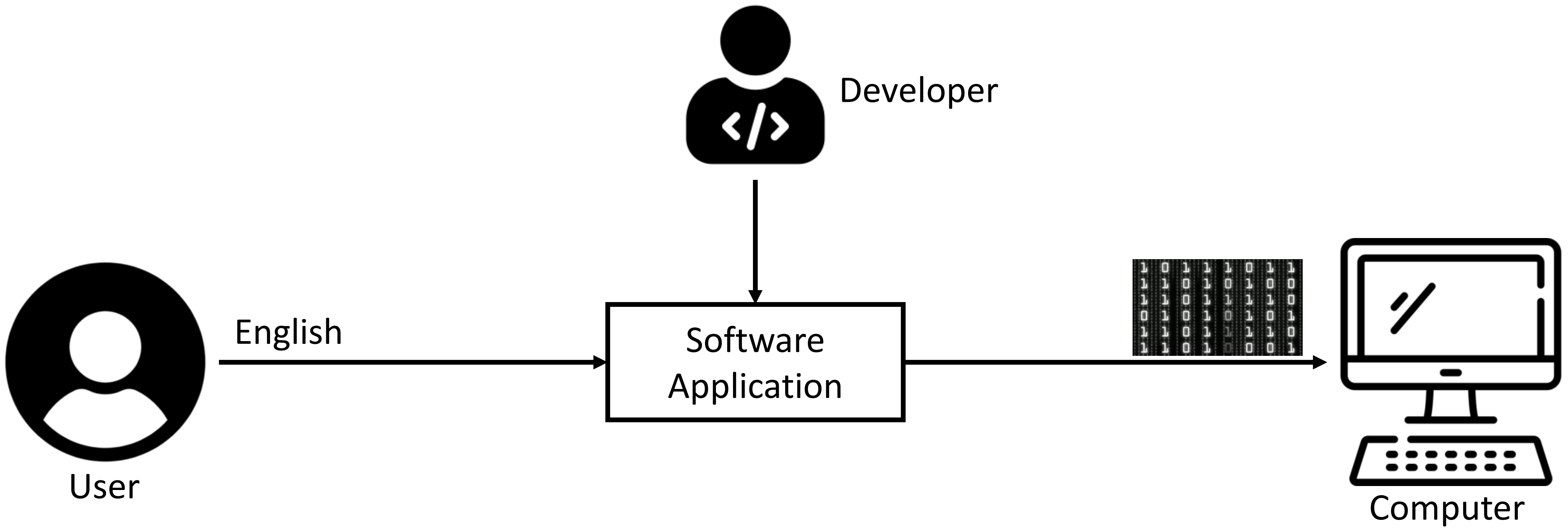
Computer Interface



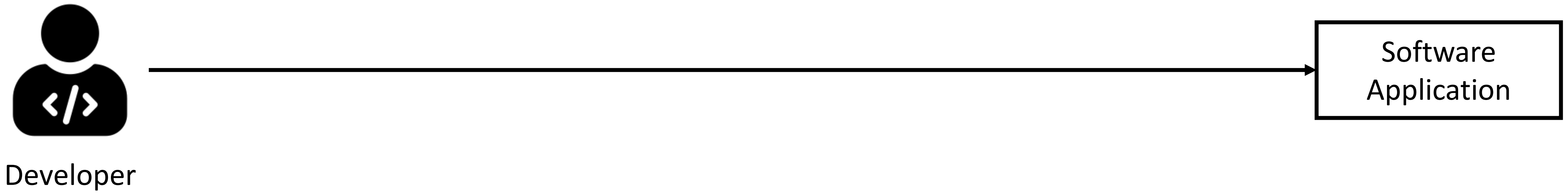
Computer Interface



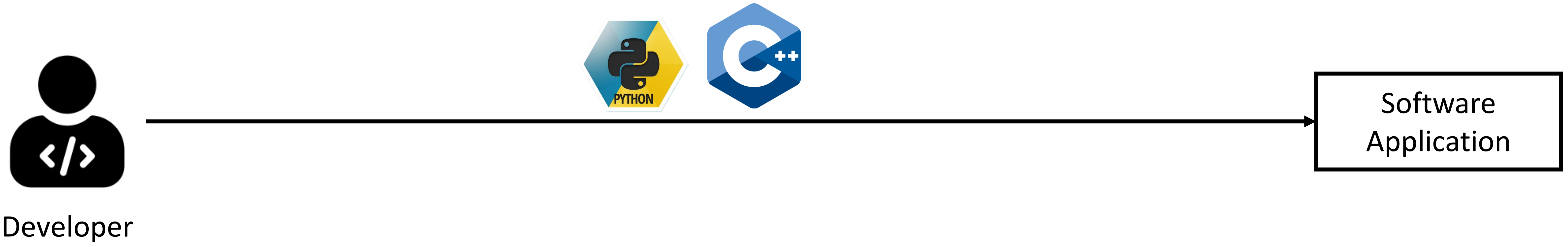
Computer Interface



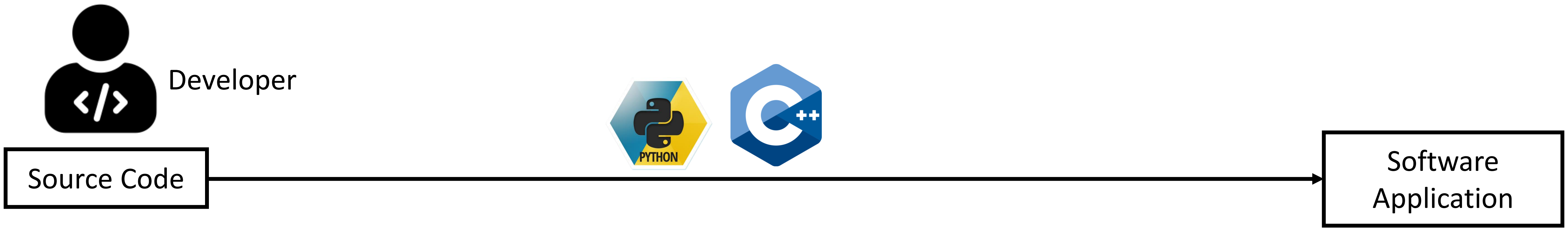
Building Software Applications



Building Software Applications



Building Software Applications



Compiled vs Interpreted Languages

- In a compiled language:

$$\begin{tikzcd}\boxed{\text{Source Code}} \arrow[r, "\text{Compiler}"] & \boxed{\text{Software Application}}\end{tikzcd}$$

Compiled vs Interpreted Languages

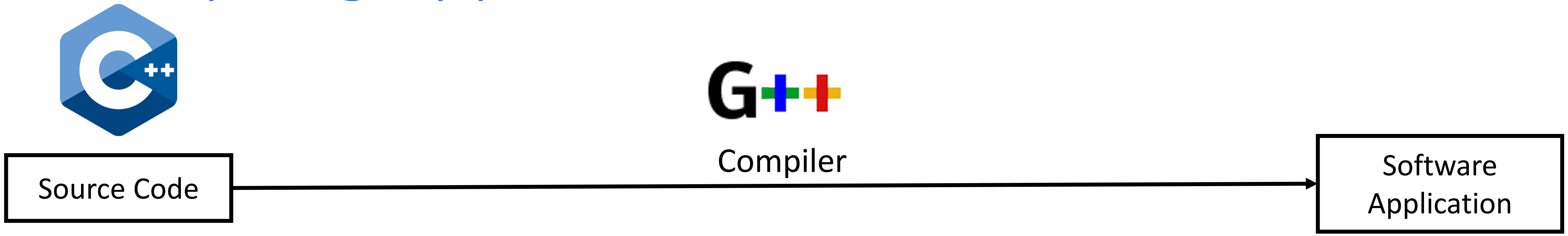
- In a compiled language:

$$\begin{tikzcd}\boxed{\text{Source Code}} \arrow[r, "\text{Compiler}"] & \boxed{\text{Software Application}}\end{tikzcd}$$

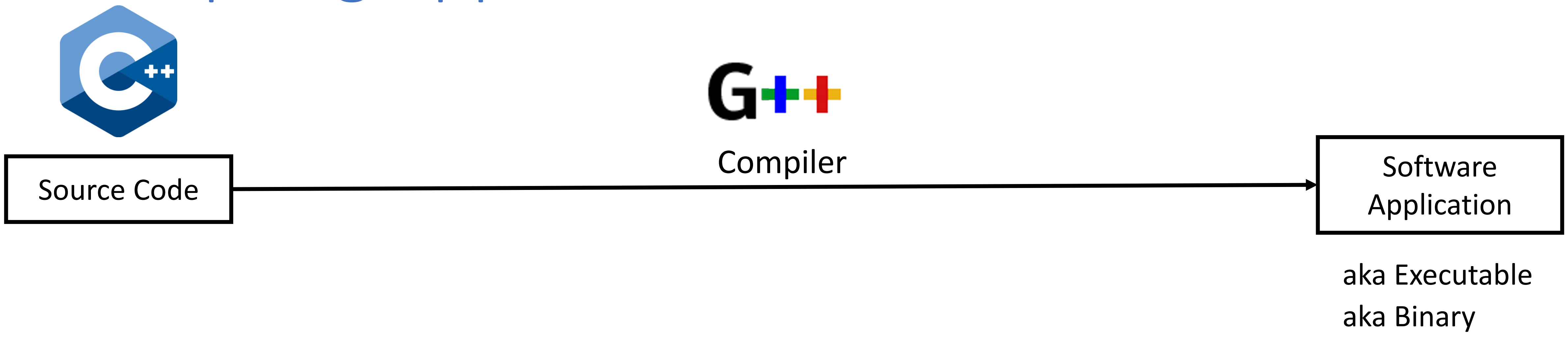
- In an interpreted language

$$\begin{tikzcd}\boxed{\text{Source Code}} \arrow[r, "\text{Interpreter}"] & \text{Implementation}\end{tikzcd}$$

Compiling Applications



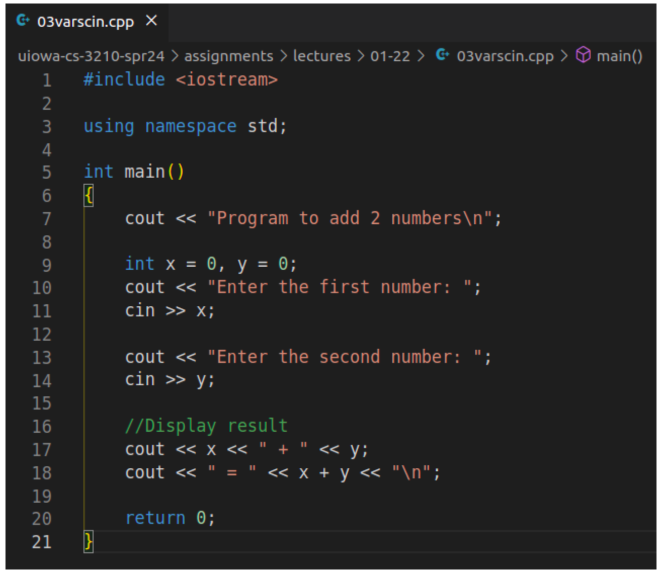
Compiling Applications



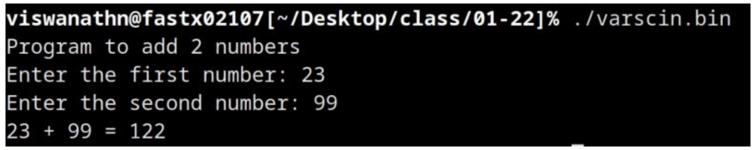
Compiling Applications

$$\begin{tikzcd}\boxed{\text{Source Code}} \arrow[r, "\text{Compiler}"] & \boxed{\text{Software Application}}\end{tikzcd}$$

```
g++ 03varscin.cpp -o varscin.bin
```

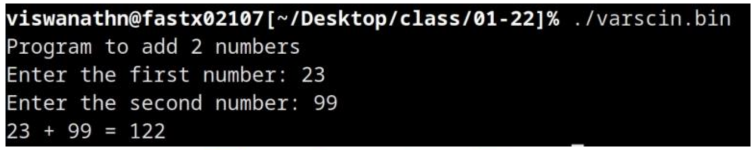

03varscin.cpp
uiowa-cs-3210-spr24 > assignments > lectures > 01-22 > 03varscin.cpp > main()

```cpp
#include <iostream>

using namespace std;

int main()
{
    cout << "Program to add 2 numbers\n";

    int x = 0, y = 0;
    cout << "Enter the first number: ";
    cin >> x;

    cout << "Enter the second number: ";
    cin >> y;

    //Display result
    cout << x << " + " << y;
    cout << " = " << x + y << "\n";

    return 0;
}
```

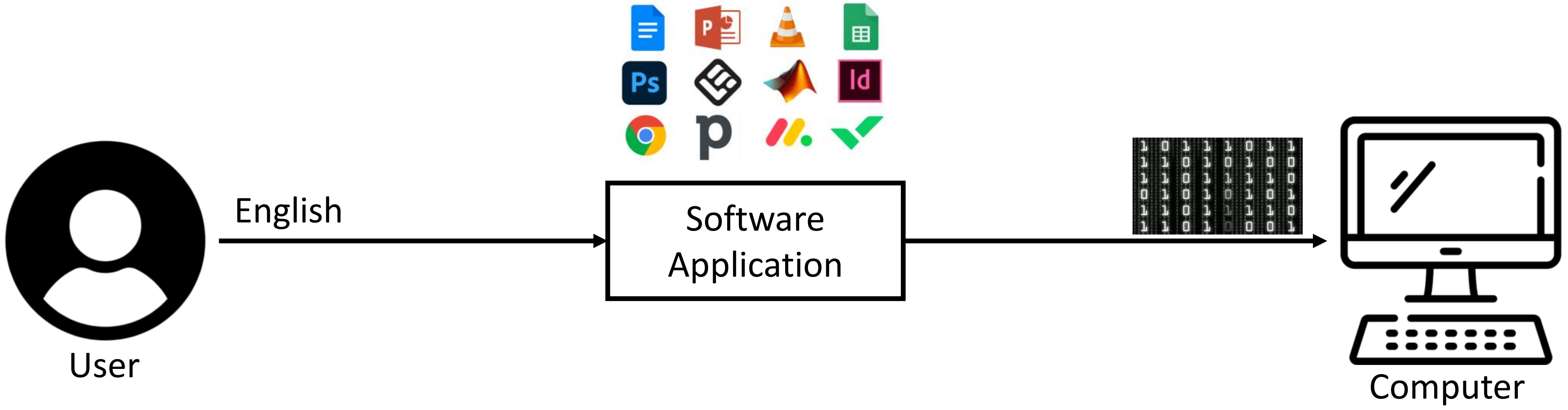


```
viswanathn@fastx02107[~/Desktop/class/01-22]% ./varscin.bin
Program to add 2 numbers
Enter the first number: 23
Enter the second number: 99
23 + 99 = 122
```

Computer Interface



```
viswanathn@fastx02107[~/Desktop/class/01-22]% ./varscin.bin
Program to add 2 numbers
Enter the first number: 23
Enter the second number: 99
23 + 99 = 122
```



Compiling Applications

1. Write your code
2. Save it
3. Compile it
   ```
   g++ filename.cpp -o filename.bin
   ```
4. Execute
   ```
   ./filename.bin
   ```

using namespace std;

VSCode

Global and Local Scope

VSCode

Types

- Types define the possible values that entities such as variables can take
- Informal description of common C++ types:

Type	Description	Example
`bool`	true/false	`true`
`char`	Single characters enclosed in single quotes	`'a'`
`int`	–2,147,483,648 to 2,147,483,647 (typically)	`5`
`float`	Decimal numbers	`4.52`
`string`	Strings enclosed in double quotes	`"blah"`

Constants

- Constants in a C++ program are unchangeable values
- Declaring a variable as const:
  ```
  const type-name constant-name = value;
  ```
- Things that I have been calling values are called literal constants since they are unchangeable
- Examples of literal constants:
  ```
  true, 78, 1.23, ‘x’, “some string”
  ```
- Preprocessor macros can be used to define constants:
  ```
  #define roottwo 1.41
  ```

Reserved Words

asm	else	new	this
auto	enum	operator	throw
bool	explicit	private	true
break	export	protected	try
case	extern	public	typedef
catch	false	register	typeid
char	float	reinterpret_cast	typename
class	for	return	union
const	friend	short	unsigned
constexpr	goto	signed	using
continue	if	sizeof	virtual
default	inline	static	void
delete	int	static_cast	volatile
do	long	struct	wchar_t
double	mutable	switch	while
dynamic_cast	namespace	template	

In addition, the following words are reserved:

and	bitor	not_eq	xor
and_eq	compl	or	xor_eq
bitand	not	or_eq	

Statements

- A program is composed of statements, each ending with a semi-colon (;)
- Blocks or compound statements are multiple statements grouped in braces ({ … })

```
statement;
\\block:
{
   statement1;
   statement2;
}
```

Operators

Assignment (=)

- To assign r-value to l-value, use assignment operator = (not equality):

  ```
  l-value = r-value;
  int m = 24;
  ```

- L-values usually refer to memory locations
- R-values represent the content to be stored in memory

Arithmetic

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Divide
%	Remainder