WESTERN SYDNEY
UNIVERSITY

C++ Fundamentals

ELEC1006: ENGINEERING COMPUTING

Elements of a Program

WESTERN SYDNEY UNIVERSITY

- A program is made up of
  - Keywords.
  - Programmer-defined identifiers.
  - Operators.
  - Punctuation.
  - Syntax.
  - Comments.

Keywords

WESTERN SYDNEY UNIVERSITY

- Set of 84 reserved words.
- Each keyword has special meaning in C++.
- Usually denoted in blue.
- https://www.learncpp.com/cpp-tutorial/keywords-and-naming-identifiers/

alignas (C++11)	char32_t (C++11)	enum	namespace	return	try
alignof (C++11)	class	explicit	new	short	typedef
and	compl	export	noexcept (C++11)	signed	typeid
and_eq	const	extern	not	sizeof	typename
asm	constexpr (C++11)	false	not_eq	static	union
auto	const_cast	float	nullptr (C++11)	static_assert (C++11)	unsigned
bitand	continue	for	operator	static_cast	using
bitor	decltype (C++11)	friend	or	struct	virtual
bool	default	goto	or_eq	switch	void
break	delete	if	private	template	volatile
case	do	inline	protected	this	wchar_t
catch	double	int	public	thread_local (C++11)	while
char	dynamic_cast	long	register	throw	xor
char16_t (C++11)	else	mutable	reinterpret_cast	true	xor_eq

Example - Keywords

WESTERN SYDNEY UNIVERSITY

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int hour;               //declare the variables for use.
6      double rate, pay;
7
8      // Start the user interface
9      cout << "Pay Calculation Program"
10         << "How many hours do you work each week? \n";
11     cin >> hour;
12     cout << "What is your hourly rate?  \n";
13     cin >> rate;
14
15     // calculating the weekly pay
16     pay = hour * rate;
17
18     // output the weekly pay amount on the screen.
19     cout << "Your weekly pay is: $" << pay << ".\n";
20     return(0);
21 }
22
```

Programmer Defined Identifiers

WESTERN SYDNEY UNIVERSITY

- Names made up by the programmer.
- Not part of the C++ language.
- Used to represent variables, memory locations, functions, etc.
  - Here, a variable is defined as a named computer memory location that holds data.
- https://www.learncpp.com/cpp-tutorial/keywords-and-naming-identifiers/

Example – Programmer Defined Identifiers

WESTERN SYDNEY UNIVERSITY

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int hour;                //declare the variables for use.
6      double rate, pay;
7
8      // Start the user interface
9      cout << "Pay Calculation Program"
10         << "How many hours do you work each week? \n";
11     cin >> hour;
12     cout << "What is your hourly rate?  \n";
13     cin >> rate;
14
15     // calculating the weekly pay
16     pay = hour * rate;
17
18     // output the weekly pay amount on the screen.
19     cout << "Your weekly pay is: $" << pay << ".\n";
20     return(0);
21 }
22
```

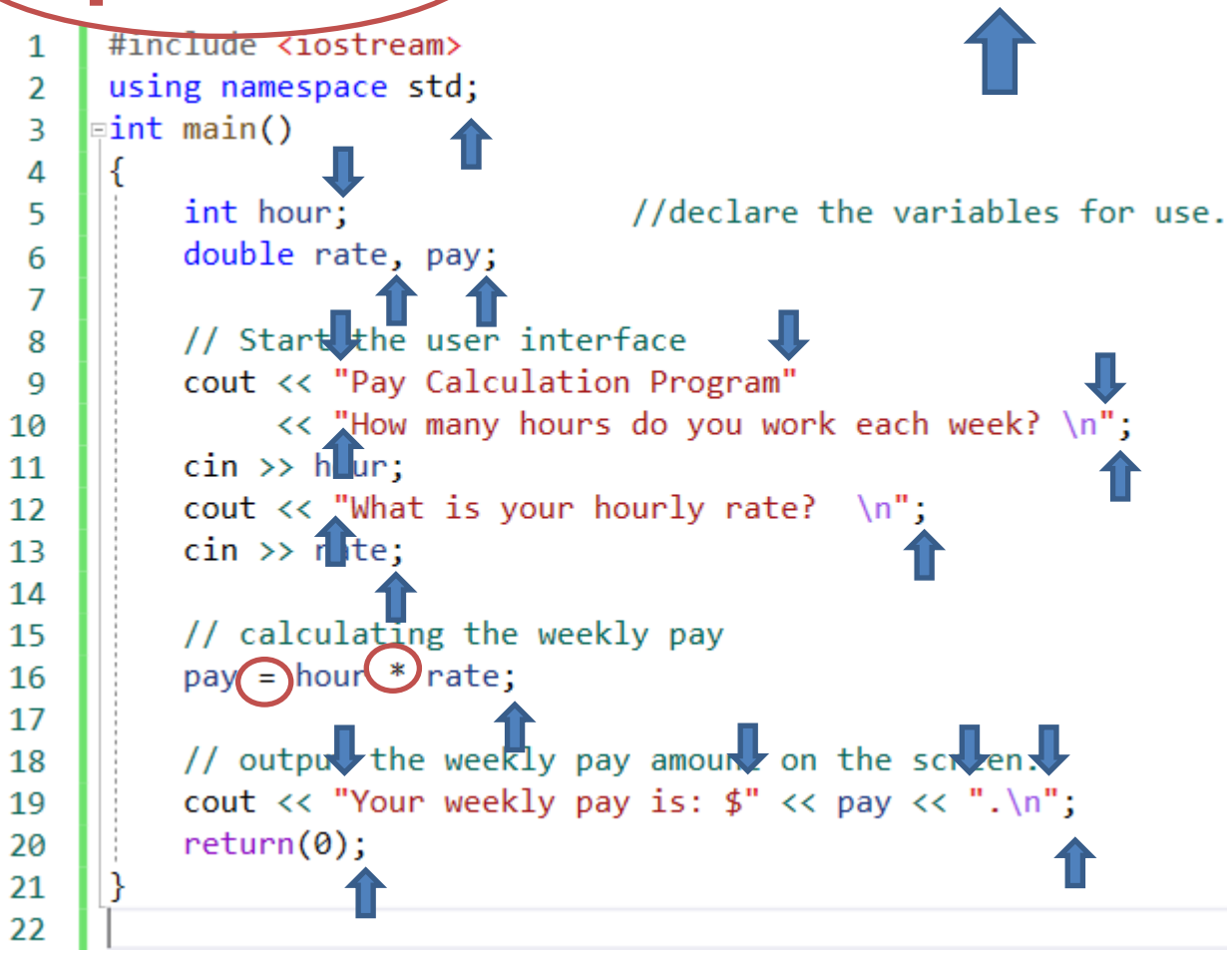
Operators & Punctuations

WESTERN SYDNEY UNIVERSITY

- Operators perform operations on data.
- Types of operators.
  - Arithmetic: +, -, *, /
  - Assignment: =
- Punctuation:
  - Mark the end of a line.
  - Separate items on a list.

Example – Operators and Punctuations

WESTERN SYDNEY UNIVERSITY



```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int hour;                  //declare the variables for use.
6      double rate, pay;
7
8      // Start the user interface
9      cout << "Pay Calculation Program"
10         << "How many hours do you work each week? \n";
11     cin >> hour;
12     cout << "What is your hourly rate?  \n";
13     cin >> rate;
14
15     // calculating the weekly pay
16     pay = hour * rate;
17
18     // output the weekly pay amount on the screen.
19     cout << "Your weekly pay is: $" << pay << ".\n";
20     return(0);
21 }
22
```

Syntax

WESTERN SYDNEY UNIVERSITY

- Rules of grammar required to be followed when writing a program.
- Controls the use of keywords, operators, programmer-defined symbols, and punctuation.

Comments

WESTERN SYDNEY UNIVERSITY

- Document parts of the source code.
- Documentation intended for persons reading the source code to:
  - Indicate the purpose of the program.
  - Describe the use of variables.
  - Explain complex sections of the code.
- Use // for single one line comment.
- Use /* … */ for multiple line comment.

Example – Syntax and Comments

WESTERN SYDNEY UNIVERSITY

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int hour;                  //declare the variables for use.
6      double rate, pay;
7
8      // Start the user interface
9      cout << "Pay Calculation Program"
10         << "How many hours do you work each week? \n";
11     cin >> hour;
12     cout << "What is your hourly rate?  \n";
13     cin >> rate;
14
15     // calculating the weekly pay
16     pay = hour * rate;
17
18     // output the weekly pay amount on the screen.
19     cout << "Your weekly pay is: $" << pay << ".\n";
20     return(0);
21 }
22
```

Ideal syntax presented for entire source code.

More info on C++

WESTERN SYDNEY UNIVERSITY

[1] Learn Cpp.com: Tutorials to help you master C++ and object-oriented programming. https://www.learncpp.com/