CHAPTER 10: ARRAYS

* Syntax: type array\_name[# element]
* Access array element: array\_name[index]
* First element has index 0
* Last element has index # element - 1
* std::array and std::vector are better than raw arrays

CHAPTER 11: POINTERS

* Syntax: type \* pointer\_name
* To make pointer points to an object (ex: int object), use & (address of): int \*p=&a
* To make pointer does not point to any object, use nullptr: int \*p=nullptr
* Access data of object pointed by pointer, use \* (dereference): int b=\*p

|  |  |
| --- | --- |
| Command | Output |
|  |  |
|  |  |
|  |  |

CHAPTER 12: REFERENCES

* Reference is an alias for another variable
* Syntax: type &ref\_name
* Changing value of alias also change value of the original
* Const-reference is read-only alias to some object
  + Syntax: const int& a=b

|  |  |
| --- | --- |
| Command | Output |
|  |  |
|  |  |

CHAPTER 13: STRINGS

* Is in C++ standard library
* Including <string> header is needed
* Create string s: std::string s = text
* Add string or character: s +=added\_text
* Access string character: [] (like array) or .at member function
* Compare string: ==
* Input string: std::cin or std::getline
* Pointer to string: .c\_str() member function
* Create substring with a starting position in original string: .substr(position, length) member function
* Find substring: .find(string\_to\_find)
  + If string is not found, return std::string::npos

|  |  |
| --- | --- |
| Command | Output |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

CHAPTER 14: AUTOMATIC TYPE DEDUCTION

* Help to automatically deduce the type of an object
* Syntax: auto c=’a’ (char type)
* Can use as part of reference type: auto& y=x
* Can use as part of constant type: const auto x=123