C++

A S R Pavan
Scientist ‘B’
NIELIT Calicut

Topics to be discussed

- Varibles
- Datatypes
- Arrays
- Vectors

Declaring and Initializing Variables

- datatype variable_name = value;
- int value; // declared but uninitialized
- int value = 200; // normal initialization
- int value (200); // constructor initialization
- int value {200}; // list initialization syntax
- int val1 = 10, val2 (20) , val3 {30}
- advantage of list initialization

Datatypes

- Datatypes
  - Primitive datatypes
    - Int, float, double, char, bool, void
  - Derived datatypes
    - array, pointer, reference,function
  - User defined
    - Structure, union, enumeration, Class

Arrays

- Collection of elements of same data type
- Continous memory
- Individual element access is possible
- Not bound check
- 1st element index is 0
- Last element index is at size-1
- Try to initialize at the time of declaration

Arrays

- Initialization
  - Datatype array_name [no_of_elemenets] {list}
  - Int marks[5]; // stores junk values
  - Int marks[5] {}; // assigns zero to all elements
  - Int marks[5] {80,85,90,95,99};
  - Int marks[5] {80,85}; // initializes 2 elem. remaining are zeros
  - Int values[2][3] {{0}};

Vectors

- Dynamic array
- Same datatype
- Continous memory
- Index starts at 0 and ends at size-1
- Provides bound checking
- Can use with lot of functions like sort, reverse, find,...
- Part of standard library std

Vectors

- Uninitialized values are zero.
- have to use #include <vector>
- Acessing using array syntax or vector syntax
  - Array syntax won’t provide bound check
  - Vector syntax provides bound check
- has methods like begin(), end(), size(), reserve()

Declaration,Initialization & Accessing

- Declaration syntax: Vector <datatype> vector_name;
  - Eg: vector <int> marks
  - Vector <int> marks {95,96,97,89}
  - Vector <int> marks (5,10) // vector size is 5 and values is 10
  - Vector <vector<int>> ratings
- Accessing vector elemnets syntax:
  - Vector_name.at(element_index)
  - Eg: marks.at(2) or marks[2]
  - Eg: ratings[2][3] or ratings.at[2].at[3]

Adding elements

- Assignment: marks.at(2) = 99;
- Adding elements using push_back method
- Adds at the end of the vector
- Syntax: vector_name.push_back(element)
  - Eg: marks.push_back(95);
  - vector_2D.pushback(vector1)

removing elements

- removing elements using pop_back()
- Removes at the end of the vector
- Syntax: vector_name.pop_back()
  - Eg: marks.pop_back();

Doubts

Q&A

End of the session

Thank You