C++

A S R Pavan
Scientist ‘B’
NIELIT Calicut

Topics to be discussed

- Decision making
- Iteration
- continue & break
- Strings
  - C-style strings
  - C++ strings

Decision making

- If
- If-else statement
- Nested if statements
- switch statement
- conditional operator/ternary operator (exp? stat1:stat2)

Switch statement

```
Switch(ctrl_expr){
    case expr1:     block1;         break;
    case expr2:     block1;         break;
                ...
                ...
    case expr3:     blockn;         break;
    default:        default_block;
}
```

- Ctrl_expr must evaluate to an integer type or enumeration type.
- Ctrl_expr value compared with case expr values

Iteration

- For loop : specific no. of times
- Range-based for loop : based on range or collection
- While loop : iterates until the condition is true
- Do-while loop : iterates until the condition is true with 1 iteraction by default
- Continue :
- break statements

Range based for loop

- Based on acessing each element without worrying about length, condition, incrementing, decrementing, subscripting indexes
- Introduced in c++11
- Syntax: for(var_type var_name:collection){

  block of statements;

  }

continue & break

- Continue:
  - Skips the next statements in the loop
  - Control goes to the beginning of the loop for next iteration

- Break:
  - Skips the next statements in the loop
  - Terminates the loop
  - Control goes to the next statement after loop construct

Strings

- C++ supports 2 types of strings
  - C-style strings
  - C++ strings
- C-style strings
  - Continous in memory
  - Terminated with null
  - As an array of characters or string literal

Strings

- c-style strings
  - char my_str [8] {“hello”}
  - We can assign elements using array style assignment
  - Eg: my_str[3] = ‘p’
- We have to include <cstring>
- Functions that work with c-style strings
  - Copying, concatenation, comparison, ...

Strings

- C++ strings
  - #include <string>
  - Continous memory
  - Dynamic size
  - Work with input and output streams
  - Operators like +,=,<,<=,>,>=,+=,==,..... can be used
  - Always initialized to empty string during declaration
  - Can use assignment operator
    - Eg: string s1 = {“Hai”};

Strings

- C++ strings
  - Accessing characters using array style [] or at() method
  - Eg: string s1 ={“Hai”}; cout << s1.at(1) << endl; // a or s1[1]
  - Objects are compared character by character
    - Can compare std::string objects
    - Can compare std::string object and c-style string literal
    - Can compare std::string object and c-style string variable
  - substr, erase, clear, length,.... methods can be used
  - getline(cin,string_name)

Doubts

Q&A

End of the session

Thank You