Programming Methodology I COEN-243 Section N NA

SEVENTH TUTORIAL

November 02, 2022

Functions Overloading

- Functions are most useful when they are made modular
- Functions overloading: One function name that can have different return types, arguments and implementations
- Overloading is only feasible when the re-defined function is distinct
- Alternatively related (not in scope of the course): *Templates*
- https://cplusplus.com/doc/tutorial/functions2/

PBV vs PBR

- **Arguments/Parameters can either be pass-by-value or pass-by-reference**
- **Pass-by-value:** the argument makes a copy of the passed variable value and uses it
  - The value of the passed variable is unaffected outside the function
- **Pass-by-reference:** the argument becomes a reference to the passed variable
  - The value of the passed variable can change accordingly
- In PBV only the value is provided while in PBR the memory location is provided
- https://www.educative.io/answers/pass-by-value-vs-pass-by-reference

Function calls

- **A function defined globally can be called anywhere**
- In competitive programming, one can make new functions from multiple other functions
- **Activity:** Double triangle
  - Function 1:
    - A) *type* upperTriangle(string str);
    - B) *type* upperTriangle(char arr[]);
  - Function 2:
    - A) *type* lowerTriangle(string str);
    - B) *type lowerTriangle(char arr[])*
  - Function 3: *type* doubleTriangle(string strU, string strL, char c)

Default Values & Static Storage

In programming, user behaviour is unexpected

- So, it can be a good practice to provide default cases
- In functions, parameters can have default values
- E.G., int divide(int a, int b=2); void printS(string str, int count=1);

The 'static' keyword can provide tracker variables in functions

- If a variable is set static in a function, its value is stored after each function call
- **Activity:** void printCalls();

Object-Oriented Programming (OOP)

- OOP is about objects that contain both data and functions *(W3Schools)*
- Procedural Programming vs OOP
- OOP Advantages:
  - It provides a better and clearly structured code
  - It is faster and easier to execute
  - It keeps C++ code DRY (Don't Repeat Yourself) – easier to maintain, modify, and debug
  - It forms the basis for multiple data structures (ways to store data)
  - It provides desired flexibility

Classes & Objects

- In OOP, objects belong to classes
- ‘Class’ in programming comes from the word ‘Classification’
- Classes provide the attributes and methods for certain *type of objects*
- Objects (similar to variables) can be classified to one class but they can have different homogenous attributes, a.k.a. characteristics
- **E.G.**, class of cars, class of students, …
- https://www.w3schools.com/cpp/cpp_oop.asp

References

- https://cplusplus.com/doc/tutorial/functions2/
- https://www.educative.io/answers/pass-by-value-vs-pass-by-reference
- https://www.w3schools.com/cpp/cpp_oop.asp
- **https://github.com/TheBarzani/COEN243_Fall2022**

THANK YOU ☺