Object Oriented Analysis and Design using Java

UE19CS353

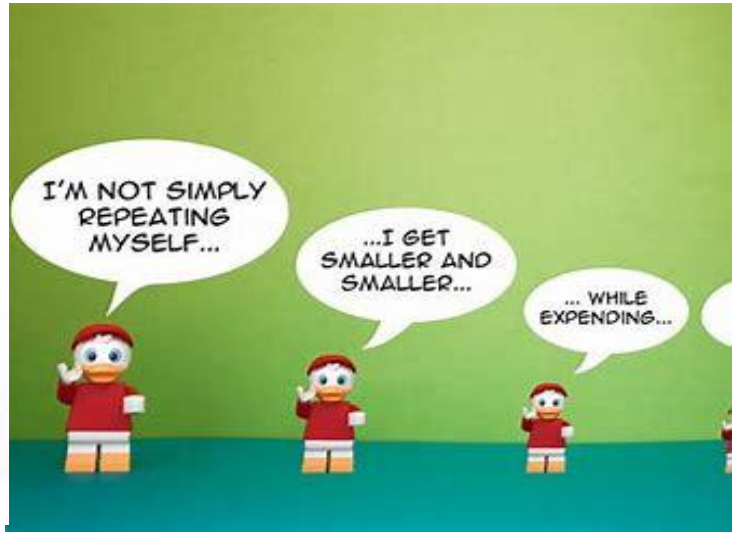
Prof. Mahitha G and Prof. Sindhu R Pai

Department of Computer Science and Engineering

Acknowledgements: Significant portions of the information in the slide sets presented through the course in the class, are extracted from the prescribed text books, information from the Internet. Since these are only intended for presentation for teaching within PESU, there was no explicit permission solicited. We would like to sincerely thank and acknowledge that the credit/rights remain with the original authors/creators only

UE19CS353: Object Oriented Analysis and Design using Java

Topic??



Prof. Mahitha G and Prof. Sindhu R Pai

Department of Computer Science and Engineering

Agenda

- Introduction
- Memory Location
- Halting Condition
- Basic rules of recursion
- Coding examples – Demo
- Points to think!
- References

Introduction

- Process which comes into existence when a function calls a copy of itself to work on a smaller problem

- Provides a way to break complicated problems down into simple problems which are easier to solve.

- Separate memory is maintained at each recursive call.

- Occurs when a thing is defined in terms of itself or of its type

Memory Allocation

- Any method which calls itself is called recursive method, and such function calls are called recursive calls
- Each recursive call creates a new copy of that method in the memory
- Once some data is returned by the method, the copy is removed from the memory
- A separate **stack is maintained at each recursive call** to store variables and other stuff declared inside method
- Once the value is returned from the corresponding method, the stack gets destroyed
- Recursion involves complexity in resolving and tracking the values at each recursive call

Halting condition

- Recursive functions can run into the problem of infinite recursion.

- Infinite recursion is when the function never stops calling itself.

- Every recursive function should have a halting condition, which is the condition where the function stops calling itself.

Basic rules of recursion

Base cases: You must always have some base or trivial case, which can be solved without recursion.

Making progress: For the cases that are to be solved recursively, the recursive call must always be to a case that makes progress toward the base case.

Design rule: Assume that all the recursive calls work. Use proof by induction.

Compound Interest Rule: Never duplicate work by solving the same instance of a problem in separate recursive calls. If possible, use dynamic programming.

Coding Examples

- Classical example of recursion is finding the factorial of a given number
- How to find the sum of the digits of a given number using recursion?
- Print all the numbers/symbols from n to 0.
- Add all the numbers upto n from 0.
- Towers of Hanoi

Points to think!!

- Can a static method be overloaded?
- Can a class be static?
- Can I have a class inside another class?
- When you have a static method inside one class, how do you call it in main() method which is in another class ?
- Can I have an object of one class inside another class as an instance variable?

References

- Recursion in C – javatpoint
- Dynamic Programming - GeeksforGeeks

THANK YOU

Mahitha G and Sindhu R Pai
Department of Computer Science and Engineering

mahithag@pes.edu

sindhurpai@pes.edu
+918277606459