

# CMSC 132:

## Object-Oriented Programming II



## Callbacks

---

Department of Computer Science  
University of Maryland, College Park

# Callback

- **Definition**

- Executable code passed as argument to other code
- Examples:
  - Timer
  - Function that prints table and has as parameters a number of rows and a second function that defines the data to print (e.g., square roots, factorials, etc.)

- **Approach**

- Higher-level code passes function as argument to lower-level code
  - In Java, pass object implementing interface
  - In C/C++, pass pointer to function
- Lower-level code invokes callback function to perform desired task

# Callback

- **Motivation**

- Keeps code separate
  - Clean division between higher & lower-level code
- Promotes code reuse
  - Lower-level code supports different callbacks
- Supports event-driven programming
  - Lower-level code registers function as handler

- **Other Examples**

- Observer pattern → `Observer.update( )`
- Visitor pattern → `Visitor.visit( )`
- A function that prints a formatted table for values computed with different functions