

# Introduction to Resource Management Solutions

- What is meant by "resources"?
  - A resource is an object which gives access to a feature from the program's environment
    - Heap memory
    - Disk file
    - Database connection
    - Window handle
- Why do resources need to be managed?
  - Typically resources are limited and should be released when no longer needed

- What is meant by a "garbage collector"?
  - A part of the program's runtime environment which monitors resource usage. When a resource is no longer needed, the garbage collector will release it
- Suggest a reason why C++ does not support garbage collection by default
  - Overhead
  - Non-deterministic - the programmer has no control over when a resource is released

- Give some examples of classes from the C++ Standard Library which manage resources
  - `std::file`, `std::string`, `std::vector`
- What is meant by "encapsulation" in the context of C++ classes?
  - The internal details of the class are hidden from code which uses the class. For example, we don't need to know how `std::file` manages disk files

- What is meant by RAII?
  - A C++ idiom for managing resources in which
    - The resource is stored as a private member of the class
    - The class constructor takes ownership of the resource
    - The class's public member functions control access to the resource
    - The class's destructor releases the resource

- Describe some of the advantages of RAII
  - Simplifies resource management (encapsulation)
    - To acquire the resource, create an instance of the RAII class. If the resource is not available, the instance creation fails
    - To use the resource, call member functions of the RAII class. These will correctly transfer ownership of the resource, if necessary
    - To release the resource, destroy the instance (or let it go out of scope). The resource will be released deterministically