Qt File Manipulation Exam

Exam Structure

- 5 progressive exercises (30 minutes each)
- Covers text files, binary files, directories, and error handling
- Solutions involve both code and conceptual understanding

1 Exercise 1: Basic File Operations (30 minutes)

1. What are the main Qt classes used for file operations?
2. Write a Qt program that:
   (a) Creates a text file named ”notes.txt”
   (b) Writes ”Hello, Qt File Handling!” into it
   (c) Closes the file properly
3. How would you check if the file was successfully created?
4. What is the difference between ‘QFile‘ and ‘QTextStream‘?
5. How can you ensure that file resources are properly released?

2 Exercise 2: Reading and Appending to Files (30 minutes)

1. Write a Qt function that reads the content of ”notes.txt” and displays it in ‘qDebug()‘.
2. Modify the function to append a new line (”This is a new line.”) to the file.
3. What happens if the file does not exist? How would you handle this error?
4. Explain the difference between:
   - ‘QIODevice::ReadOnly‘
   - ‘QIODevice::WriteOnly‘
   - ‘QIODevice::Append‘
5. How would you read a file line by line instead of all at once?

3 Exercise 3: Working with Directories (30 minutes)

1. What Qt class is used for directory operations?
2. Write a program that:
   (a) Checks if a directory named ”QtFiles” exists in the current folder
   (b) Creates it if it doesn’t exist
   (c) Lists all files inside it
3. How would you copy a file from one directory to another using Qt?
4. What is the difference between ‘QDir::entryList()‘ and ‘QDir::entryInfoList()‘?
5. How can you check file permissions (read/write/execute) in Qt?

4 Exercise 4: Binary Files and Serialization (30 minutes)

1. What is the difference between text and binary files?
2. Write a program that:
   (a) Saves a list of integers ‘10, 20, 30, 40, 50‘ into a binary file ”data.bin”
   (b) Reads them back and prints them
3. How would you store and retrieve a custom ‘struct‘ (e.g., ‘Person‘) in a binary file?
4. What is ‘QDataStream‘, and why is it useful for binary files?
5. How would you handle endianness issues when reading binary files across platforms?

5 Exercise 5: Error Handling and Best Practices (30 minutes)

1. What are common file operation errors, and how can you handle them in Qt?
2. How would you implement a safe file deletion with error checking?
3. What is the best way to handle file paths in a cross-platform Qt application?
4. Why should you avoid hardcoding file paths? What alternatives exist?
5. How would you implement a simple logging system that writes to a file?

Learning Outcomes

After completing this exam, students should be able to:

- Read and write text/binary files in Qt
- Work with directories and file systems
- Handle errors and edge cases in file operations
- Apply best practices for file management in Qt applications