Name: Utkarsh Vinayak Kshirsagar

Reg.No: 2022BIT031

Subject: Software Engineering

ASSIGMENT-1

Report on C, C++, Java: Documentation, Libraries, Frameworks, Development Tools, IDEs, and Graphics Programming

C Programming Language

1. Documentation:

- Official Documentation: The official documentation for C is provided by ANSI and ISO standards. The latest standard documentation is the ISO/IEC 9899.

- Books: "The C Programming Language" by Brian W. Kernighan and Dennis M. Ritchie is the most authoritative book.

- Online Resources: Websites like cprogramming.com and tutorials on platforms like GeeksforGeeks and W3Schools.

2. Libraries:

- Standard Library: The C Standard Library provides macros, type definitions, and functions for tasks like input/output, memory management, and string manipulation.

- Third-Party Libraries: Examples include GLib for data structures and utilities, and OpenSSL for cryptographic functions.

3. Frameworks:

- C does not have frameworks in the same way as higher-level languages but relies on libraries for extended functionality.

4. Development Tools:

- Compilers: GCC (GNU Compiler Collection), Clang.

- Debuggers: GDB (GNU Debugger).

- Build Systems: Make, CMake, Ninja.

5. IDEs:

- Code::Blocks: A free, cross-platform IDE that supports multiple compilers.

- Eclipse CDT: An open-source IDE for C and C++.

- Visual Studio: Provides extensive support for C/C++ development with powerful debugging and code analysis tools.

6. Graphics Programming:

- Libraries:SDL (Simple DirectMedia Layer), OpenGL.

- Frameworks: Allegro, SFML (Simple and Fast Multimedia Library).

C++ Programming Language

1. Documentation:

- Official Documentation: Provided by ISO standards, the latest being ISO/IEC 14882.

- Books: "The C++ Programming Language" by Bjarne Stroustrup.

- Online Resources: cppreference.com, learncpp.com.

2. Libraries:

- Standard Library: Includes the Standard Template Library (STL) for data structures and algorithms.

- Boost: A collection of peer-reviewed, portable C++ source libraries.

- Qt: A library for GUI development and cross-platform applications.

3. Frameworks:

- Qt Framework: For cross-platform GUI applications.

- Boost Framework: For various functionalities like file system, threading, and networking.

4. Development Tools:

- Compilers: GCC, Clang, MSVC (Microsoft Visual C++).

- Debuggers: GDB.

- Build Systems: CMake, Make, Ninja.

5. IDEs:

- CLion: A powerful IDE from JetBrains tailored for C and C++.

- Visual Studio: Offers robust support for C++ with advanced features.

- Qt Creator:Specifically designed for development with the Qt framework.

6. Graphics Programming:

- Libraries:SFML, SDL, OpenGL.

- Qt: Provides extensive tools for 2D/3D graphics and GUI development.

Java Programming Language

1. Documentation:

- Official Documentation: Available from Oracle’s official Java documentation website.

- Books: "Effective Java" by Joshua Bloch, "Java: The Complete Reference" by Herbert Schildt.

- Online Resources: Oracle’s Java Tutorials, GeeksforGeeks, JavaTpoint.

2. Libraries:

- Standard Library: Provides core libraries for collections, I/O, networking, and more.

- Apache Commons: A collection of libraries for various purposes.

- Google Guava: Enhances the Java standard libraries.

3. Frameworks:

- Spring Framework: For building enterprise-level applications.

- Hibernate: For ORM (Object-Relational Mapping).

- Apache Struts: For building web applications.

4. Development Tools:

- Compilers: Javac (part of the JDK).

- Debuggers: JDB (Java Debugger).

- Build Systems: Maven, Gradle, Ant.

5. IDEs:

- IntelliJ IDEA: A powerful and feature-rich IDE.

- Eclipse: An open-source IDE with a vast plugin ecosystem.

- NetBeans: Provides strong support for Java development.

6. Graphics Programming:

- JavaFX: For rich client applications.

- AWT/Swing: For creating GUI applications.

Qt Library for GUI Application Development

1. Introduction to Qt:

- Overview: Qt is a free and open-source widget toolkit for creating graphical user interfaces as well as cross-platform applications.

- Features: Rich set of GUI controls, robust graphics backend, internationalization support, and more.

2. Qt Development Tools:

- Qt Creator: The official IDE for Qt development, offering advanced features for GUI development, debugging, and profiling.

- QMake: The build system for Qt applications.

3. Qt Libraries:

- Core Modules: For non-GUI functionality like file handling, data structures, and threading.

- GUI Modules:For creating and managing windows, dialogs, and various widgets.

4. Development Process:

- Setting Up: Install Qt and Qt Creator.

- Creating a Project: Use Qt Creator to set up a new project with necessary configurations.

- Designing the GUI: Utilize Qt Designer to drag-and-drop widgets and design the interface visually.

- Coding the Logic:Write the necessary C++ code to handle user interactions and application logic.

Calculator Application with GUI using Qt

1. Project Setup:

- Install Qt and Qt Creator: Ensure Qt is installed along with its development tools.

- Create a New Project: Open Qt Creator and start a new Qt Widgets Application project.

2. Designing the GUI:

- Open Qt Designer: Use the drag-and-drop interface to design the calculator layout.

- Widgets: Add buttons for digits (0-9), operations (+, -, *, /), and a display area (QLineEdit) for showing results.

3. Implementing Functionality:

- Signals and Slots: Connect button signals to slots for handling clicks.

- Logic Implementation: Write the necessary C++ code to perform arithmetic operations and update the display.

4. Building and Running:

- Build the Project: Use Qt Creator to build the project.

- Run the Application: Execute the application to see the calculator in action.