مقام معظم رهبری:

علم برای یک ملّت مهم‌ترین ابزار آبرو و پیشرفت و اقتدار است.

۱۳۹۴/۰۸/۲۰

Qt Training in C++

Lecturer: Ali Panahi

Autumn 1401

Introduction to Qt

What is Qt ?

Qt is cross-platform software for creating graphical user interfaces as well as cross-platform applications that run on various software and hardware platforms such as Linux, Windows, macOS, Android or embedded systems with little or no change in the underlying codebase while still being a native application with native capabilities and speed.

Qt is currently being developed by The Qt Company, a publicly listed company, and the Qt Project under open-source governance, involving individual developers and organizations working to advance Qt. Qt is available under both commercial licenses and open-source GPL 2.0, GPL 3.0, and LGPL 3.0 licenses.

Introduction to Qt

Why use Qt ?

- Design and develop great user experiences
- Qt saves you money
- Get your products to market faster
- Performance, delivered
- One framework, fewer dependencies
- Develop for any platform
- We speak many languages
- Flexible. Reliable. Qt.
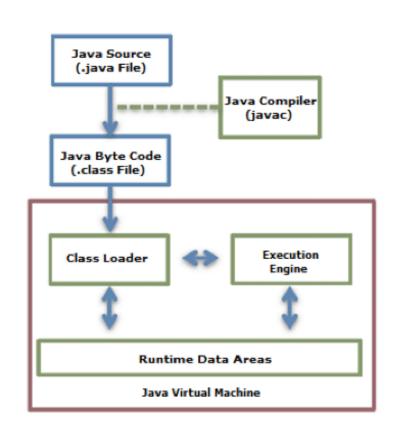- Open source and future-proof

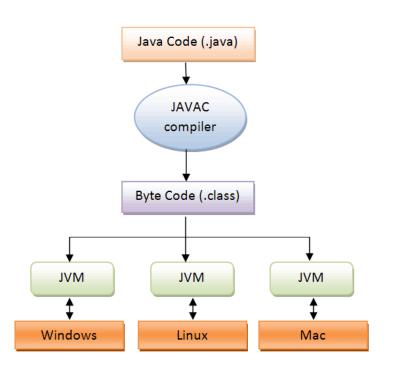## Java vs .Net vs C++ vs QT vs .Net Core

- Java
  - Java is object-oriented programming language
  - Java is a general-purpose
  - Java is cross-platform
  - Java is class-based
  - Java is managed
  - Java is free
  - Low execution speed
  - Disassembling object code
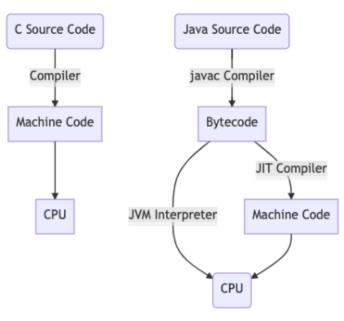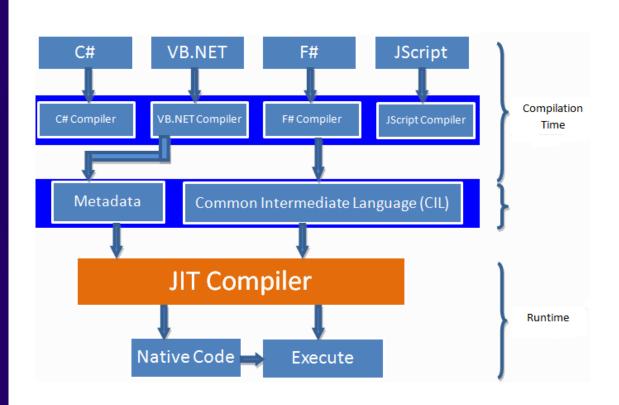 - High development speed

Java vs .Net vs C++ vs QT vs .Net Core

- .Net
  - Multi-language Support
  - .Net is managed
  - Automatic resource management
  - .Net is commercial (Microsoft)
  - Low execution speed
  - Disassembling object code
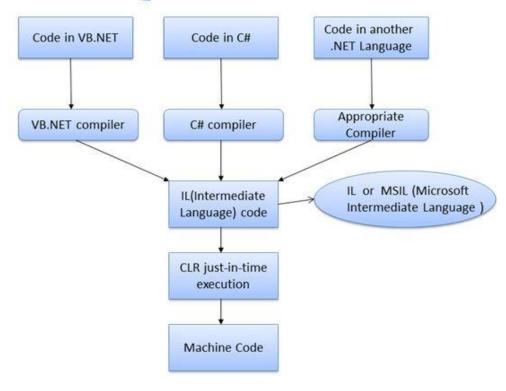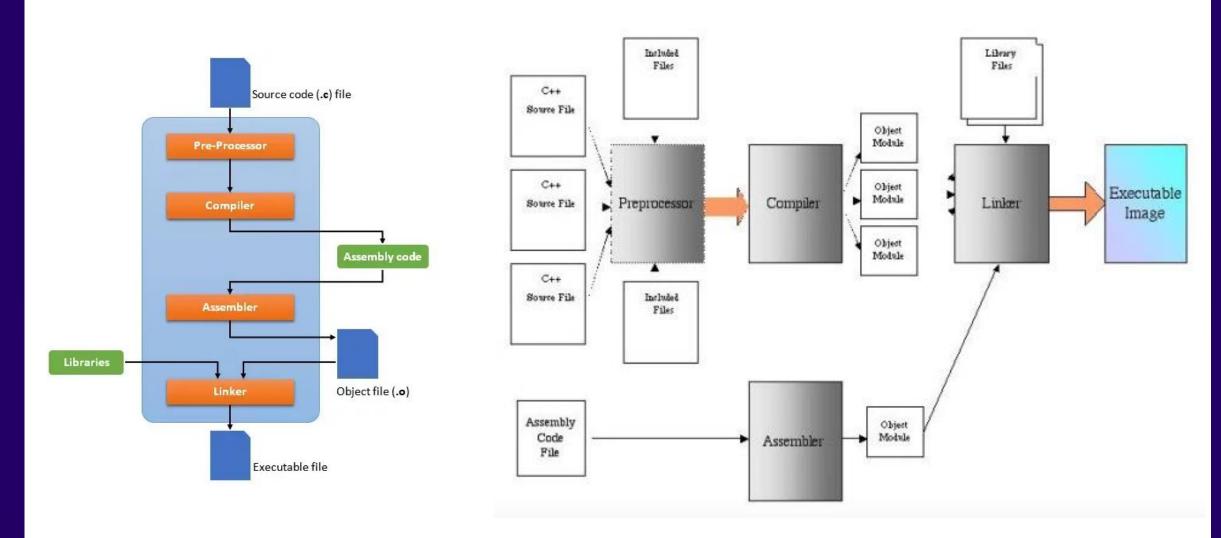 - Only for windows
  - High development speed

Java vs .Net vs C++ vs QT vs .Net Core

- C/C++
  - It is native
  - Creating libraries that can be used in other languages
  - Powerful & fast
  - High security
  - Platform dependent
  - Irreversibility and reverse engineering of codes
  - It has no memory management
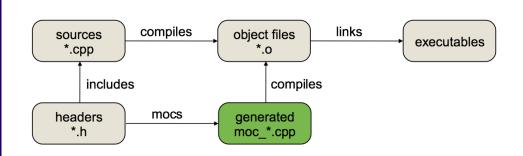 - Low development speed

Java vs .Net vs C++ vs QT vs .Net Core
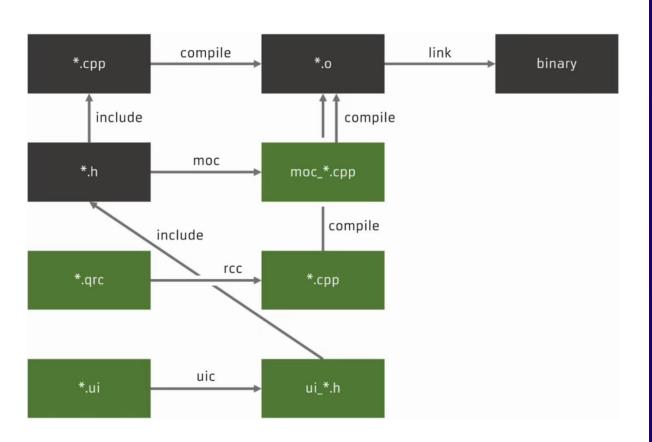
- Qt
  - It is native
  - High security
  - Cross-platform (Windows, Linux, Mac)
  - Many possibilities
  - Free and open source
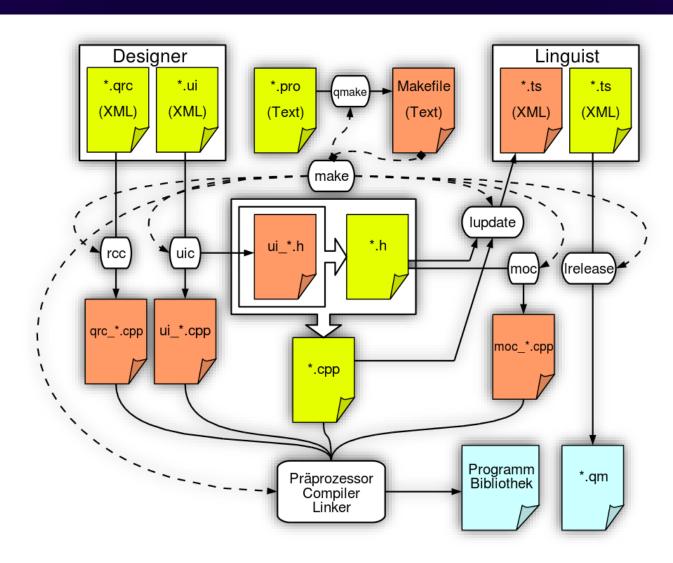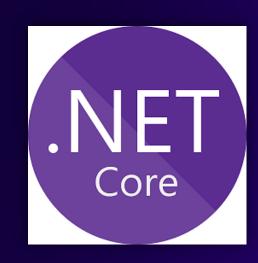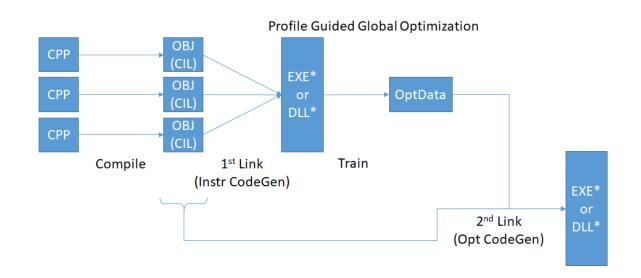
Java vs .Net vs C++ vs QT vs .Net Core
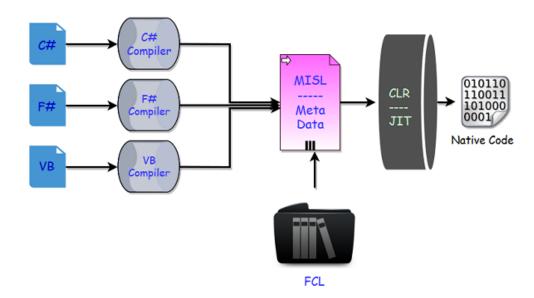
- .Net Core
  - Multi-language support
  - .Net core is managed
  - Automatic resource management
  - .Net core is free and open source (Microsoft)
  - Low execution speed
  - Disassembling object code
  - Cross-platform (Windows, Linux)
  - High development speed

Profile Guided Global Optimization

CPP → OBJ (CIL)
CPP → OBJ (CIL)
CPP → OBJ (CIL)

Compile

EXE* or DLL*

1st Link (Instr CodeGen)

Train

OptData

2nd Link (Opt CodeGen)

EXE* or DLL*

C# → C# Compiler
F# → F# Compiler
VB → VB Compiler

MISL ----- Meta Data

CLR ---- JIT

010110 110011 101000 0001
Native Code

FCL

Companies using Qt

Qt helps the best companies in the world deliver better user experiences faster.

Panasonic	**HARMAN**	**AMD**
In-flight entertainment systems	In-vehicle infotainment system	Graphics software
medec	**ZUKEN**	**DENSO**
Anesthesia & critical care medical devices	EDA & CAD end-to-end engineering solutions	Automotive mobility technology

Introduction to Qt   >>   Qt Versions

Version	Release date / Support until	Target (Windows)
Qt 0.90	1995	-
Qt 1.0	1996	-
Qt 2.0	1999	-
Qt 3.0	2001	-
Qt 4.0	2005	-
Qt 4.8 LTS (4.8.7)	2011	XP, 7, 8.1, 10
Qt 5.6 LTS (5.6.3)	2016 / 2019	-
Qt 5.9 LTS (5.9.9)	2017 / 2020	-
Qt 5.12 LTS (5.12.12)	2018 / 2021	-
Qt 5.15 LTS	2020 / 2025	7, 8.1, 10, 11
Qt 6.2 LTS (6.2.6)	2021 / 2024	10, 11
Qt 6.4	2022 / 2023	10, 11

Introduction to Qt

Languages that support the Qt library

Qt is developed with C++ language.

Qt can be used in several programming languages other than C++, such as Python, Javascript, C# and Rust via language bindings; many languages have bindings for Qt 5 and bindings for Qt 4.

Qt compilers

- Windows
  - MSVC
  - MinGW (Gnu Based)
  - Cygwin (BSD Based)
- Linux
  - GCC/G++
  - CLang

Introduction to Qt

Qt development environments

- Qt Creator
  - Windows
  - Linux
- Visual Studio
  - Windows
- PyCharm (Only for PyQt)
  - Windows
  - Linux

Installation methods

- From installer
  - Offline installer
  - Online installer
    - o From Packages (Linux)
    - o From Setup wizard
- From source

Offline Installer

- It can be downloaded from the link below:
  - URL: https://www.qt.io/offline-installers
- It is precompiled
- Only available for limited editions
- No internet required
- The installer size is large

Offline Installer

On Windows

qt-opensource-windows-x86-%VERSION%.exe

On Linux

qt-opensource-linux-x64-%VERSION%.run

Online Installer

- It can be downloaded from the link below:
  - URL: https://download.qt.io/official_releases/online_installers/
- It is precompiled
- Only available for limited editions
- Internet required
- The installer size is small

Online Installer

On Windows

qt-unified-windows-x64-%VERSION%-online.exe

On Linux

qt-unified-linux-x64-%VERSION%-online.run

Name	Last modified	Size	Metadata
↑ Parent Directory		-	
qt-unified-windows-x64-online.exe	09-Nov-2022 10:52	41M	Details
qt-unified-mac-x64-online.dmg	09-Nov-2022 10:52	18M	Details
qt-unified-linux-x64-online.run	09-Nov-2022 10:52	55M	Details

For Qt Downloads, please visit qt.io/download

QtÂ® and the Qt logo is a registered trade mark of The Qt Company Ltd and is used pursuant to a license from The Qt Company Ltd.
All other trademarks are property of their respective owners.

The Qt Company Ltd, Bertel Jungin aukio D3A, 02600 Espoo, Finland. Org. Nr. 2637805-2

List of official Qt-project mirrors

Online Installer

On Windows

Run "qt-unified-windows-x64-%VERSION%-online.exe" as administrator

On Linux

This prerequisite must be installed in the Debian operating system

```
sudo apt-get install libxcb-xinerama0
```

A file with the name qt-unified-linux-x-online.run will be downloaded, then add exec permission.

```
chmod +x qt-unified-linux-x-online.run
```

Remember to change 'x' for the actual version of the installer. Then run the installer.

```
./qt-unified-linux-x-online.run
```

Offline/Online Installer

Install Qt in any operative system

Current sample is 4.5.0 version.

Once you've downloaded Qt and opened the installer program, the installation procedure is the same for all operative systems, although the screenshots might look a bit different. The screenshots provided here are from Linux.

Login with a existing Qt account or create a new one:

Offline/Online Installer
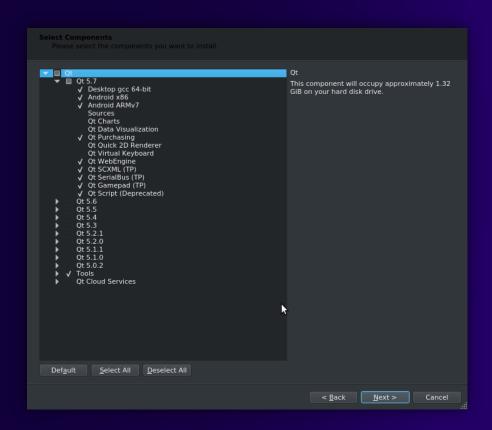
Install Qt in any operative system

Select a path to install the Qt libraries and tools

Installation Folder

Please specify the folder where Qt will be installed.

/home/imuser/Qt   Browse...

< Back   Next >   Cancel

Offline/Online Installer

Install Qt in any operative system

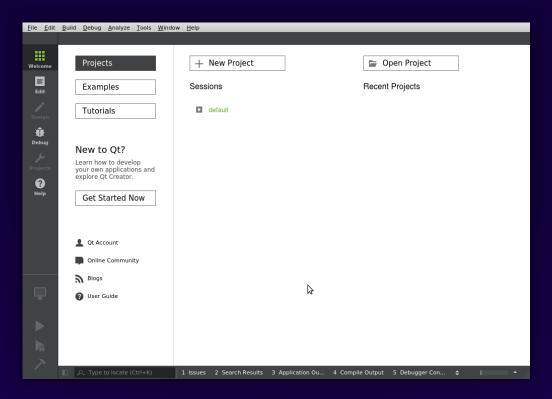Select the library version and the features you want

Offline/Online Installer

Install Qt in any operative system

After downloading and the installation is finished, go to the Qt installation directory and launch Qt Creator or run it directly from the command line.

Installation From Packages (Linux)

Installation in Debin (11.x) distribution

```
$ sudo apt-get update && sudo apt-get upgrade
$ sudo apt install qtbase5-dev qt5-qmake qtbase5-dev-tools
$ sudo apt-get install qtcreator
$ qtcreator
```
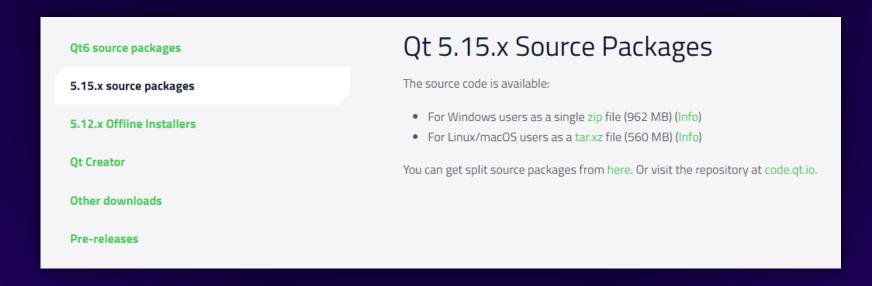
Type of build

- Dynamic
  - Dynamic plug-in is basically a shared library which is loaded at runtime.
    - ✓ Ability to update and patch Qt libraries.
    - ✓ The Qt libraries should be included with the final executable file
- Static
  - Static plug-in is built into your executable (like a static lib).
    - ✓ The libraries are combined with the executable file and a final file is created
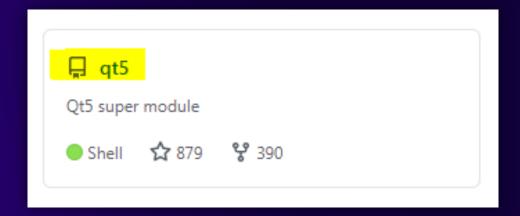
Build in Windows

- It can be downloaded from the link below (windows version 5.15.2):
    - URL: https://www.qt.io/offline-installers
- Ability to customize modules
- The ability to change the source code
- Ability to compile statically

Build in Windows

- Enter the following website (Qt requirements to compile):
    - URL: https://github.com/qt

System requirements

- CMake 3.18 or later
- Perl 5.8 or later
- Python 2.7 or later
- C++ compiler supporting the C++17 standard

It's recommended to have ninja 1.8 or later installed.

For other platform specific requirements, please see section "Setting up your machine" on:
http://wiki.qt.io/Get_The_Source

Build in Windows

- Prerequisites for compiling Qt on the windows platform

Windows:

1. Open a command prompt.
2. Ensure that the following tools can be found in the path:

- Supported compiler (Visual Studio 2019 or later, or MinGW-builds gcc 8.1 or later)
- Perl version 5.12 or later [http://www.activestate.com/activeperl/]
- Python version 2.7 or later [http://www.activestate.com/activepython/]
- Ruby version 1.9.3 or later [http://rubyinstaller.org/]

```
cd <path>\<source_package>
configure -prefix %CD%\qtbase
cmake --build .
```

35

Build in Windows

- Compiler:
  - Visual Studio 2019 or later
  - MinGW-builds gcc 8.1 or later
- Tools:
  - Perl version 5.12 or later [http://www.activestate.com/activeperl/]
  - Python version 2.7 or later [http://www.activestate.com/activepython/]
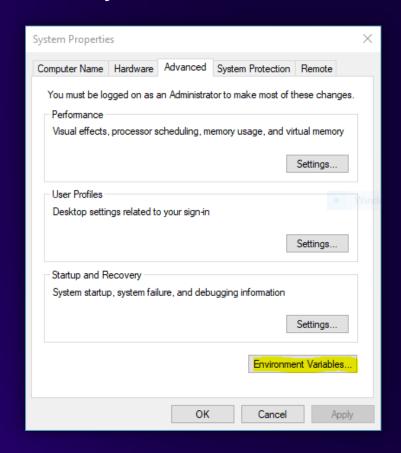 - Ruby version 1.9.3 or later [http://rubyinstaller.org/]

  Note: "Python" and "Perl" installation requires internet.

  Note: Compilation requires Windows version 7 or higher.

Build in Windows

- After installing the tools (Perl, Python, Ruby), their path should be added to the "System Enviroment"

Build in Windows

- First, download the source code

    - Example: qt-everywhere-src-5.15.2.zip

- Decompress the source code in the desired path, for example "c:\"

Build in Windows >> Compile with MSVC

- Install Visual Studio 2019
- Depending on the required architecture (32-bit or 64-bit version), we run the Visual Studio command line from start menu:

  - x86 Native Tools Command Prompt for VS 2019

  - x64 Native Tools Command Prompt for VS 2019

- Enter the following commands in the command line:

```
> SET _ROOT=C:\Qt\qt-everywhere-src-5.15.2
> SET PATH=%_ROOT%\qtbase\bin;%_ROOT%\gnuwin32\bin;%PATH%
> SET PATH=%_ROOT%\qtrepotools\bin;%PATH%
```

39

Build in Windows >> Compile with MSVC

- Enter the Qt source code path

```
> Cd C:\Qt\qt-everywhere-src-5.15.2
```

- Configure the compiler

```
> configure -debug-and-release -platform win32-msvc -developer-build
  -prefix "C:\Qt\5.15.2-x86" -nomake examples -nomake tests
  -skip qtwebengine -opensource -mp
```

- Start the compilation with the following command:

  - nmake

- install Qt with the following command

  - nmake install

- Clean source code

  - nmake clean

Thank You

By Ali Panahi