Learning C++ **Summer 2025** **Mobina Amrollahi**

Exercise 1

1. Briefly describe the design aims of C++ and comment on the extent to which C++ meets those.
   Answer. C++ was designed to combine the strengths of C as a systems programming language with Simula's facilities for organizing programs. To meet these two objectives, C++ was designed to be a *general-purpose programming language* with a bias towards *systems programming* that

   - is a better C
   - supports data abstraction
   - supports object-oriented programming
   - supports generic programming

2. What are the main programming styles supported by C++?
   Answer. C++ supports:

   - data abstraction, i.e., classes
   - object-oriented programming, i.e., inheritance and polymorphism
   - generic programming, i.e., templates

3. Check the following page: www.stroustrup.com/applications and list 20 major real-world C++ applications.
   Answer.

   (a) **Adobe Systems**: All major applications are developed in C++:
   - Photoshop & ImageReady
   - Illustrator
   - Acrobat
   - InDesign
   - GoLive
   - Frame (mostly C, some C++)

   (b) **Amazon.com**: Software for large-scale e-commerce.

   (c) **Apple**: OS X is written in a mix of languages, but a few important parts are in C++:
   - Finder
   - IOKit device drivers (IOKit is the only place where C++ is used in the kernel.)

   (d) **AT&T**: The largest US telecommunications provider.
   - 1-800 service
   - Provisioning systems
   - Systems for rapid network recovery after failure

   (e) **Autodesk**: A large number of major applications in the CAD domain.

   (f) **Bloomberg**: Provides real-time financial information to investors.

(g) **CERN**: Data analysis for large high-energy physics experiments using the ROOT toolset and libraries.

(h) **Ericsson**:

- TelORB – distributed operating system with object-oriented distributed RAM database
- Base for the TSP application server platform
- TDMA-CDMA HLR
- GSM-TDMA-CDMA mobility gateway
- AAA server

(i) **Facebook (now Meta)**: Several high-performance and high-reliability components.

(j) **Google**:

- Web search engine
- Chromium browser
- Google File System
- MapReduce large-cluster data processing

(k) **HP (Hewlett-Packard)**: A small sample of HP's C++ applications:

- C, C++, Fortran90 compilers, and linker for the HP IA64 platform (over 1M lines of C++)
- SAM (system management utility)
- Networking libraries in HP-UX
- Java VM core
- Parts of OpenView
- Non-stop XML parser (originally from Compaq)

(l) **IBM**:

- OS/400
- K42 – high-performance, open-source OS kernel for cache-coherent multiprocessors

(m) **Intel**:

- VTune performance analysis software
- Compilers and optimizers
- Various chip design and manufacturing software

(n) **Microsoft**: Most products are built using Visual C++. Major products include:

- Windows XP, Vista, 7, NT (NT4, 2000), 95, 98, Me
- Microsoft Office (Word, Excel, Access, PowerPoint, Outlook)
- Internet Explorer and Outlook Express
- Visual Studio (C++, VB, FoxPro)
- Exchange, SQL
- Minor products: FrontPage, Money, Picture It, Project
- Z3 (open-source theorem prover from Microsoft Research)
- All Microsoft games

(o) **MongoDB**: An open-source database used widely in web applications and large enterprises (e.g., Viacom, Disney).

(p) **Morgan Stanley**: C++ is used extensively for financial modeling.

(q) **Mozilla**: Firefox browser and Thunderbird mail client.

(r) **MySQL**: MySQL Server (about 250,000 lines of C++) and MySQL Cluster.

(s) **NASA**:

- Mars rover autonomous driving system (scene analysis and route planning)
- James Webb Telescope software
- International Space Station software components

(t) **Games**: *Warcraft III*, *World of Warcraft*.

Exercise 2

1. What does a compiler do? What does a linker do?
   Answer. A compiler processes the source text of C++ program, and further returns object files which are combined through a linker resulting in an executable program.

2. Change the *Hello, world!* program to output the two lines:

   ```
   Hello, programming!
   Here you go!
   ```

   Answer.

   ```
   #include <iostream>
   int main() {

              std::cout << "Hello, programming! \n";
              std::cout << "Here you go! \n";

              return 0;
              }
   ```

3. Is this a valid program? Why or why not?

   ```
   #include <iostream>
   int main() { std::cout << "Hello, world!" << std::endl; }
   ```

   Answer. Yes, this is a valid program. Since no value is returned, (also known as Void) the system interprets this program as a successful one with the main function returning nothing.

4. Is this a valid program? Why or why not?

   ```
   #include <iostream>
   int main() std::cout << "Hello, world!" << std::endl;
   ```

   Answer. No, this is not a valid program. The systems expects curly braces after any function defined, such as the "main" in this situation.

5. What is the shortest valid program?
   Answer.

```
#include <iostream>
int main() {}
```

   There's no return value from the main function to the system which is interpreted as success.