NobleProg

Lab Exercise 8– C++ Build using Bazel

(External C++ Libraries in a Bazel)

Objective: Create a lab exercise to teach the integration of External C++ Libraries in a Bazel project.

Prerequisites:

- Bazel installed on your machine.
- Basic knowledge of Bazel and C++.
- Text editor for code editing.

Scenario: You have a C++ project, and you want to integrate and use an external library (e.g., Google's Abseil library). Your goal is to set up the project, integrate the external library, and build and run the project successfully using Bazel.

Step 1: Project Setup:

Create a diretory structure for your lab exercise:

```
lab_external_dependencies_cpp/
|-- WORKSPACE
|-- BUILD
|-- main.cpp
```

Step 2: Write C++ Code:

Create a simple C++ source file for your project:

main.cpp (Executable):

```
#include <iostream>

#include "absl/strings/str_cat.h"

int main() {

   std::cout << "Using Abseil library: " << absl::StrCat("Hello, ", "world!") << std::endl;

   return 0;

}
```

Step 3: Configure the WORKSPACE and BUILD Files:

WORKSPACE (in the project root):

```
load("@bazel_tools//tools/build_defs/repo:http.bzl", "http_archive")

http_archive(

  name = "com_google_absl_new",

  urls = ["https://github.com/abseil/abseil-
cpp/archive/98eb410c93ad059f9bba1bf43f5bb916fc92a5ea.zip"],

  strip_prefix = "abseil-cpp-98eb410c93ad059f9bba1bf43f5bb916fc92a5ea",

)
```

BUILD (in the project root):

```
cc_binary(

   name = "main",

   srcs = ["main.cpp"],

   deps = [

      "@com_google_absl_new//absl/strings",

   ],
```

```
)
```

In this example:

We use http_archive in the WORKSPACE file to download and configure the Abseil library.

In the BUILD file, we define a binary target (main) that depends on the Abseil library.

Step 4: Build and Run the Project:

Open your terminal and navigate to the project directory (lab_external_dependencies_cpp).

Build the project using Bazel:

```
bazel build //:main
```

Run the executable:

```
bazel run //:main
```

Conclusion:

This lab exercise demonstrates how to integrate and use an external library (Abseil in this case) in a Bazel C++ project. It emphasizes the importance of configuring the WORKSPACE and BUILD files to specify external dependencies and shows how to build and run a project that depends on external libraries using Bazel. This exercise provides valuable hands-on experience for working with external dependencies in a Bazel-based C++ project.