**Microservices Architecture and Programming**

**Lab Practical and date** – Practical 3, 21st August 2020

**Name and Roll Number**- Het Shah, 17BIT103

**Practical Objective**- Designing gRPC based micro-service with Ballerina

**Steps Involved-**

We used python and created a GPRC based microservices- the application will answer the user query regarding the mathematical operations such as addition, subtraction, division, multiplication and square root.

**Background**

**GRPC**

In gRPC, a client application can directly call a method on a server application on a different machine as if it were a local object, making it easier for you to create distributed applications and services. As in many RPC systems, gRPC is based around the idea of defining a service, specifying the methods that can be called remotely with their parameters and return types. On the server side, the server implements this interface and runs a gRPC server to handle client calls. On the client side, the client has a stub (referred to as just a client in some languages) that provides the same methods as the server.

gRPC clients and servers can run and talk to each other in a variety of environments - from servers inside Google to your own desktop - and can be written in any of gRPC’s supported languages. So, for example, you can easily create a gRPC server in Java with clients in Go, Python, or Ruby. In addition, the latest Google APIs will have gRPC versions of their interfaces, letting you easily build Google functionality into your applications.

**How to Run**

pip install grpcio

$ pip install grpcio-tools

$ python -m grpc\_tools.protoc -I. --python\_out=. --grpc\_python\_out=. calculator-grpc.proto

**Use the above commands to compile the proto file (calculator-grpc.proto in this case) and generate the files calculator\_grpc\_pb2.py and calculator\_grpc\_pb2\_grpc.py automatically**

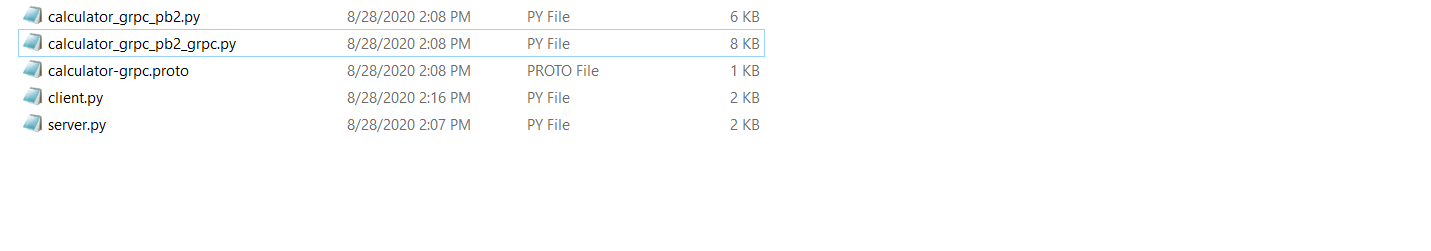
****

Figure 1 files calculator\_grpc\_pb2.py and calculator\_grpc\_pb2\_grpc.py are generated automatically

$python client.py

Open new CMD and run the following command- python client.py

$python server.py

Open new CMD window and run the following command- python server.py

Give inputs from 1 to 6 and the numbers as input to get the desired output

OUTPUT

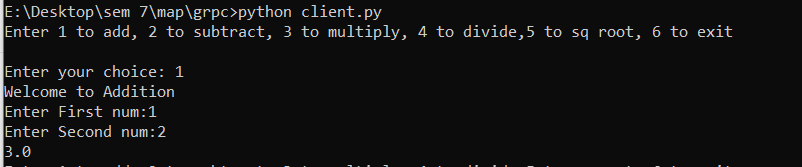


Figure 2 Option 1- Addition, adding 1 and 2 getting result 3

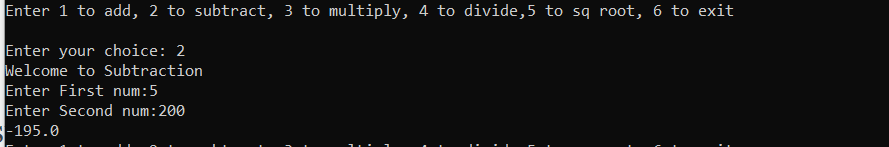


Figure 3 Option 2- subtraction, subtracting 5 and 200 getting result -195

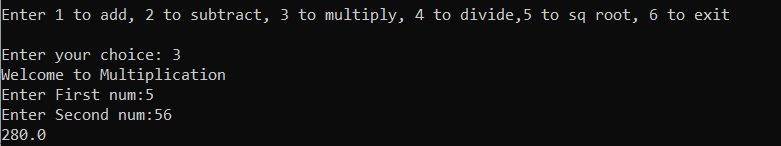


Figure 4 Option 3- multiplication, multiplying 5 and 56 getting result 280

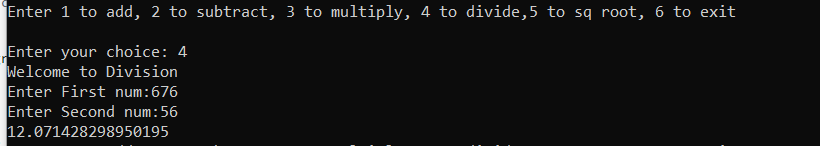


Figure 5 Option 4- division, dividing 676 and 56 getting result 12.07

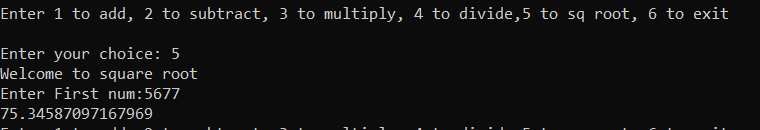


Figure 6 option 5- square root, sq root of 5677 is 75.34

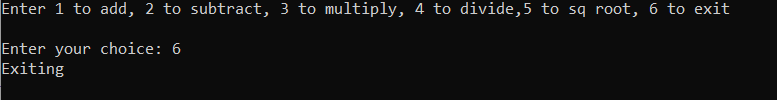


Figure 7 option 6, exit the program

**Conclusion**

In this practical we learned how to make GRPC calls to a sever using python using a protobuf file and using python for hosting the server and making the calls to find the answer of mathematical operations such as addition, subtraction, multiplication, division and square root by giving input from 1 to 5.