**Applied Distributed Systems**

**Assessment 2 – Java TCP Client Server Application**

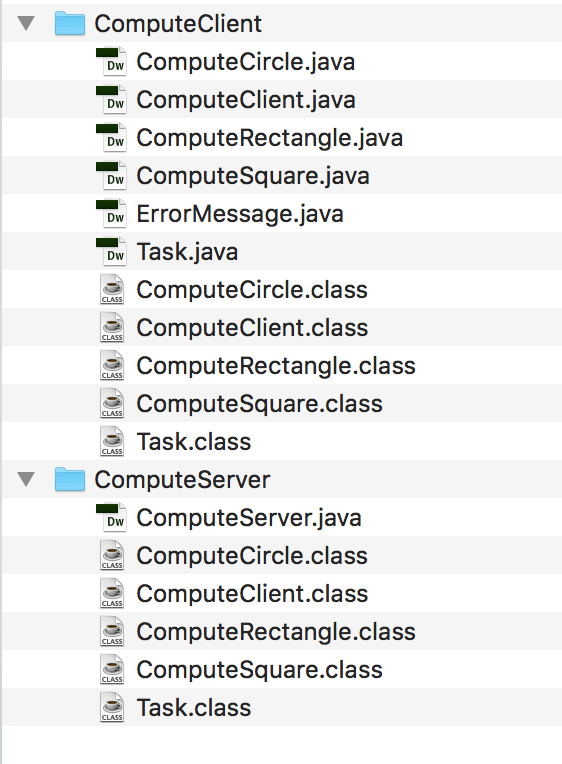
**Michael Gonzalez S0292161**

**Term 1 2018**

**Part 2 (End user instruction manual)**

**Using the program**

The client and server java files are located in two separate folders labeled **ComputeClient** and **ComputeServer** and also contain the .class files required by the server.



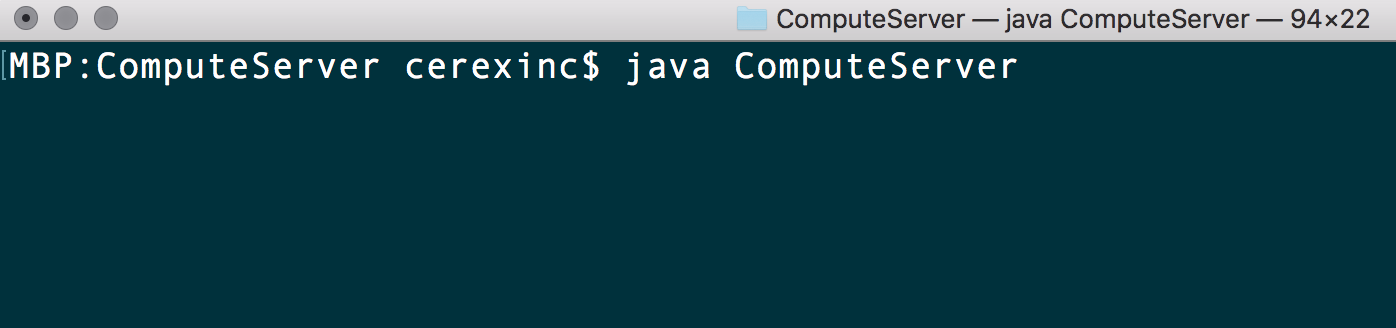
**Step One**

Before running the program, ensure the **compute-tasks (ComputeCircle, ComputeSquare, ComputeRectangle)** and **Task** **.class definitions** are located in both the **ComputeClient** and **ComputeServer** folder. If the **.class** files are not located in the **ComputeServer** folder they need to be copied and paste from the **ComputeClient** folder before running the program.

Without the .class files, the server will send an error to the client informing the user to upload the relevant .class files (**Covered in the testing section**).

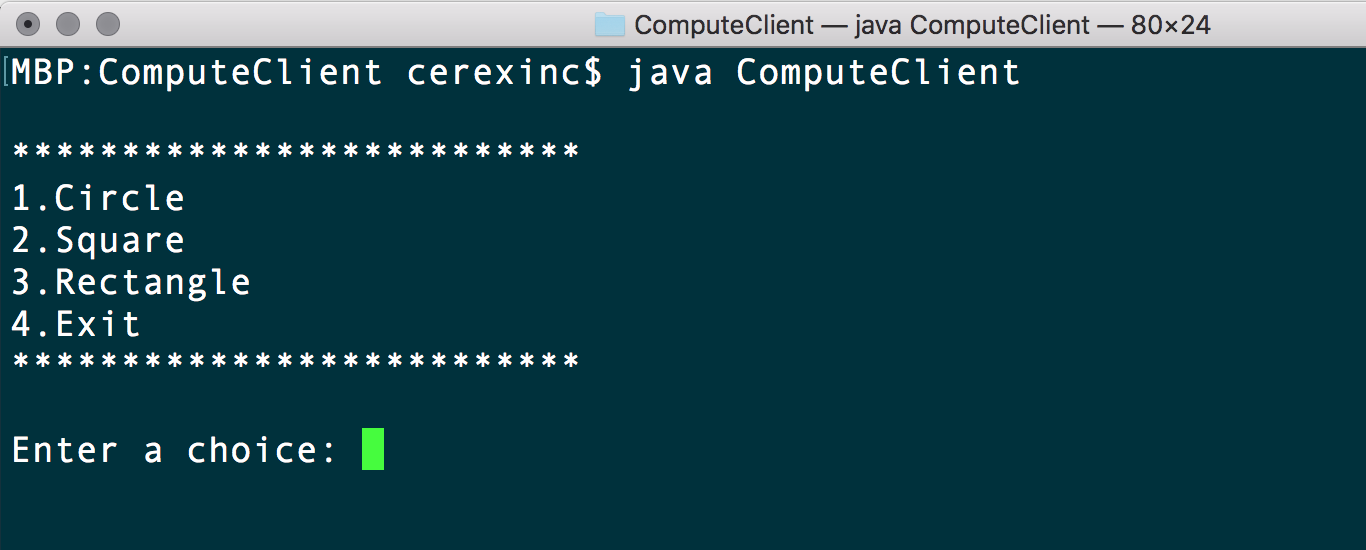
**Step Two (Running the server)**

The Server must be running first before using the program. To start the server, navigate to the **ComputeServer folder** in a terminal or command prompt and type the command **java ComputeServer** to run the server.



**Step Three (Running the client)**

With the server running, open another terminal or command prompt window and navigate to the **ComputeClient** folder. Next, type **java ComputeClient** to start the client and run the program



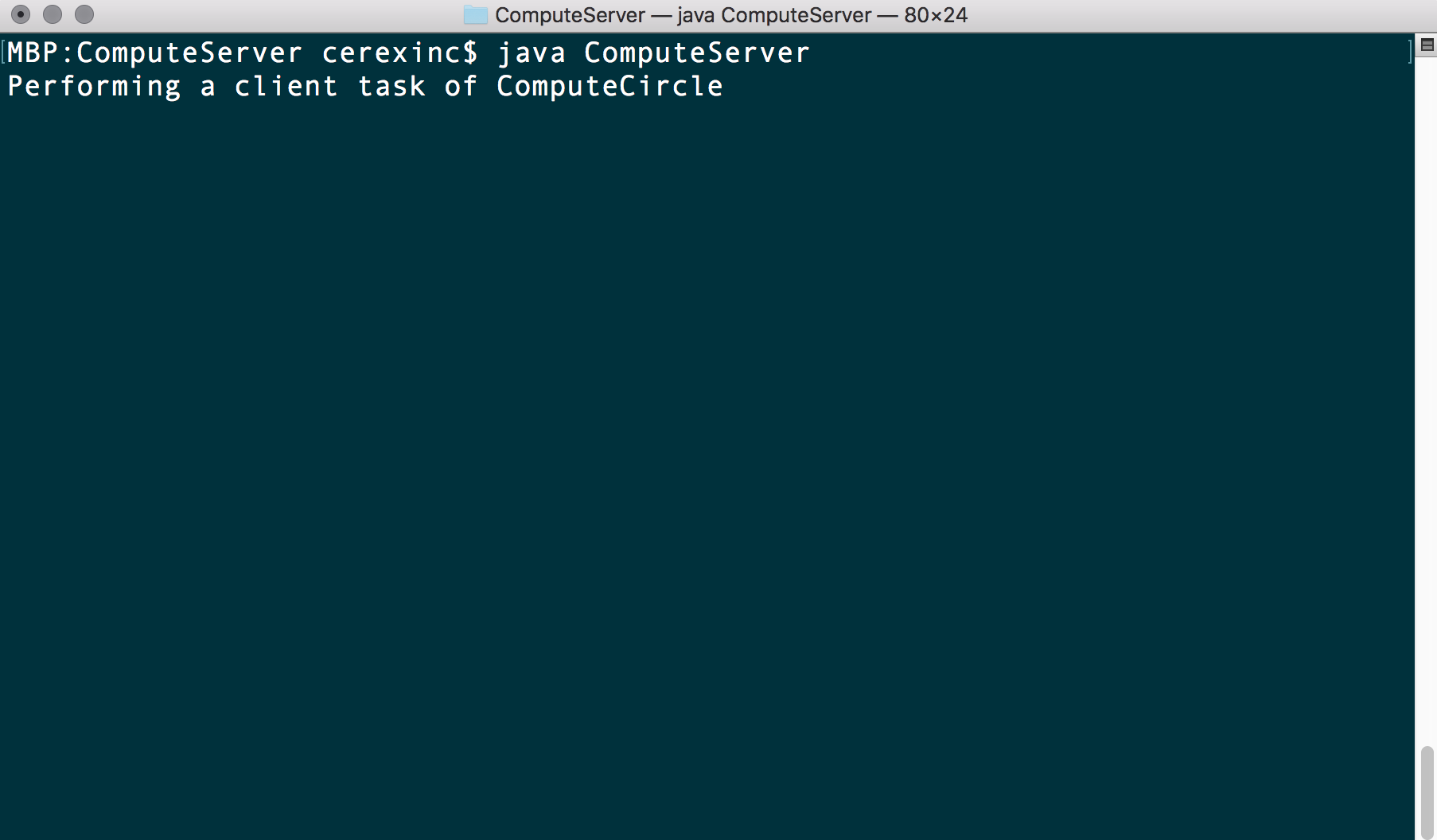
**Testing**

The following screenshots show how the client and server interact together as well as how the server handles the **ClassNotFoundException** when either of the required .class files are not located in the **ComputeServer** folder.

**Compute-task Circle**

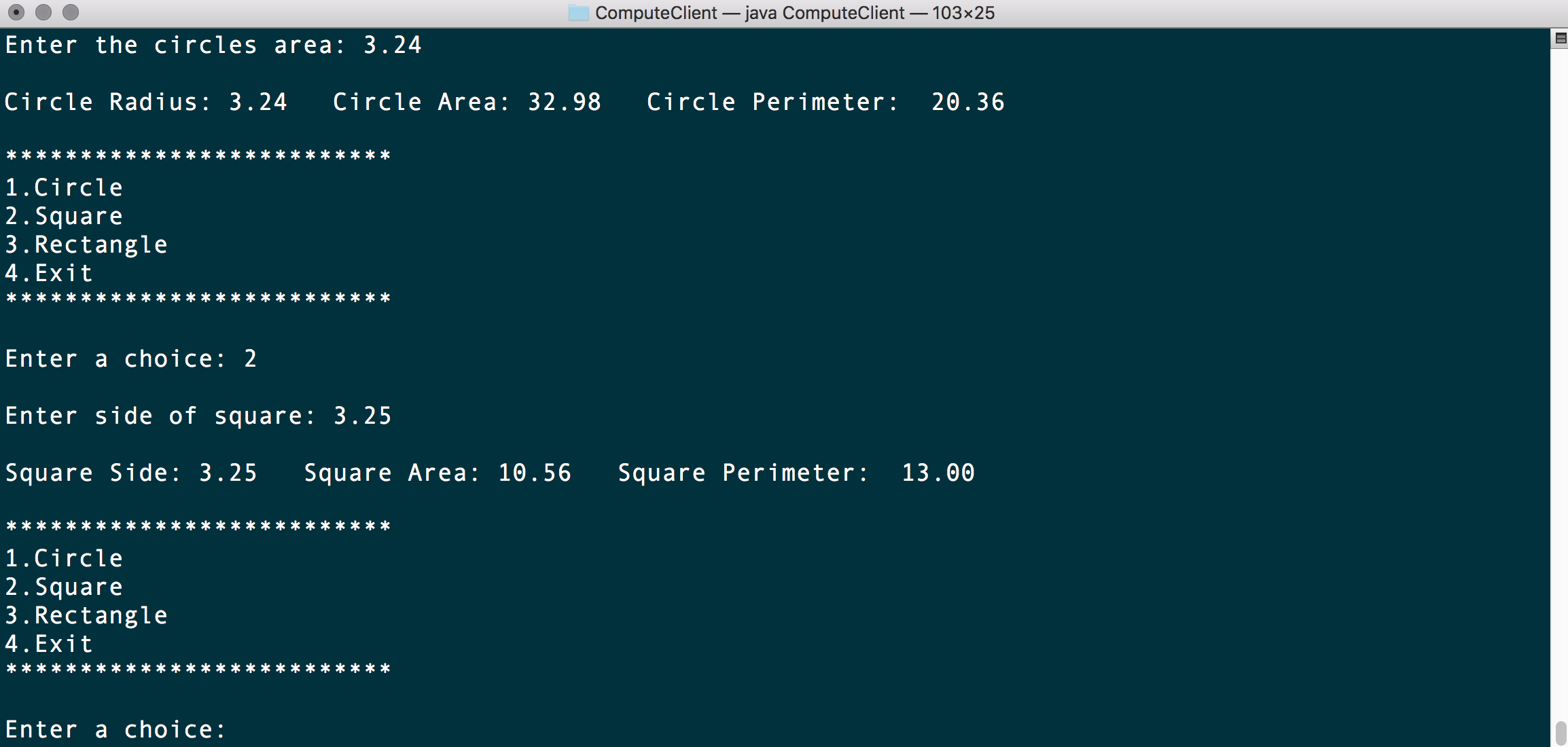
**Client Side (ComputeClient)**

****

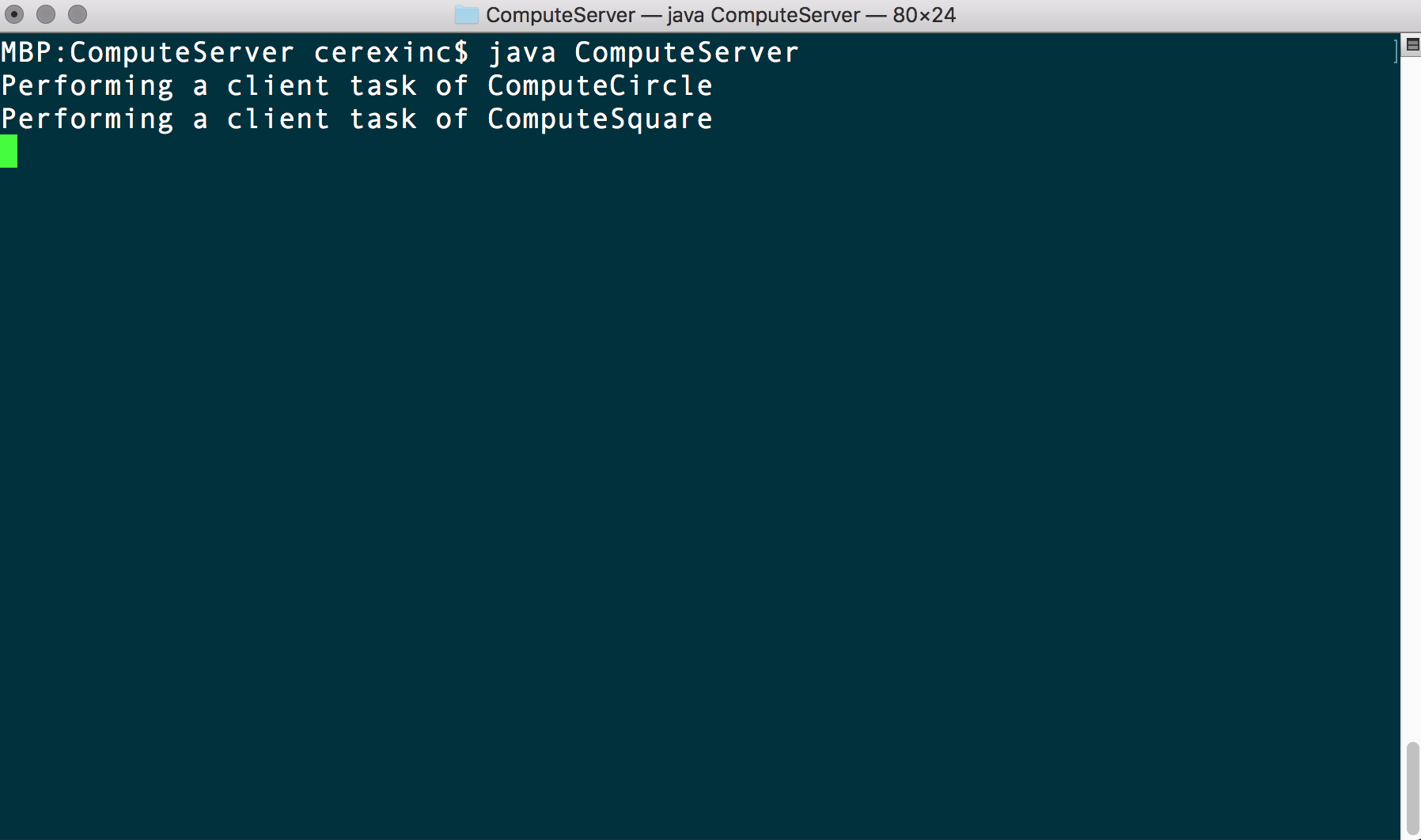
**Server Side (ComputeServer) **

**Compute-task Square**

**Client Side (ComputeClient)**

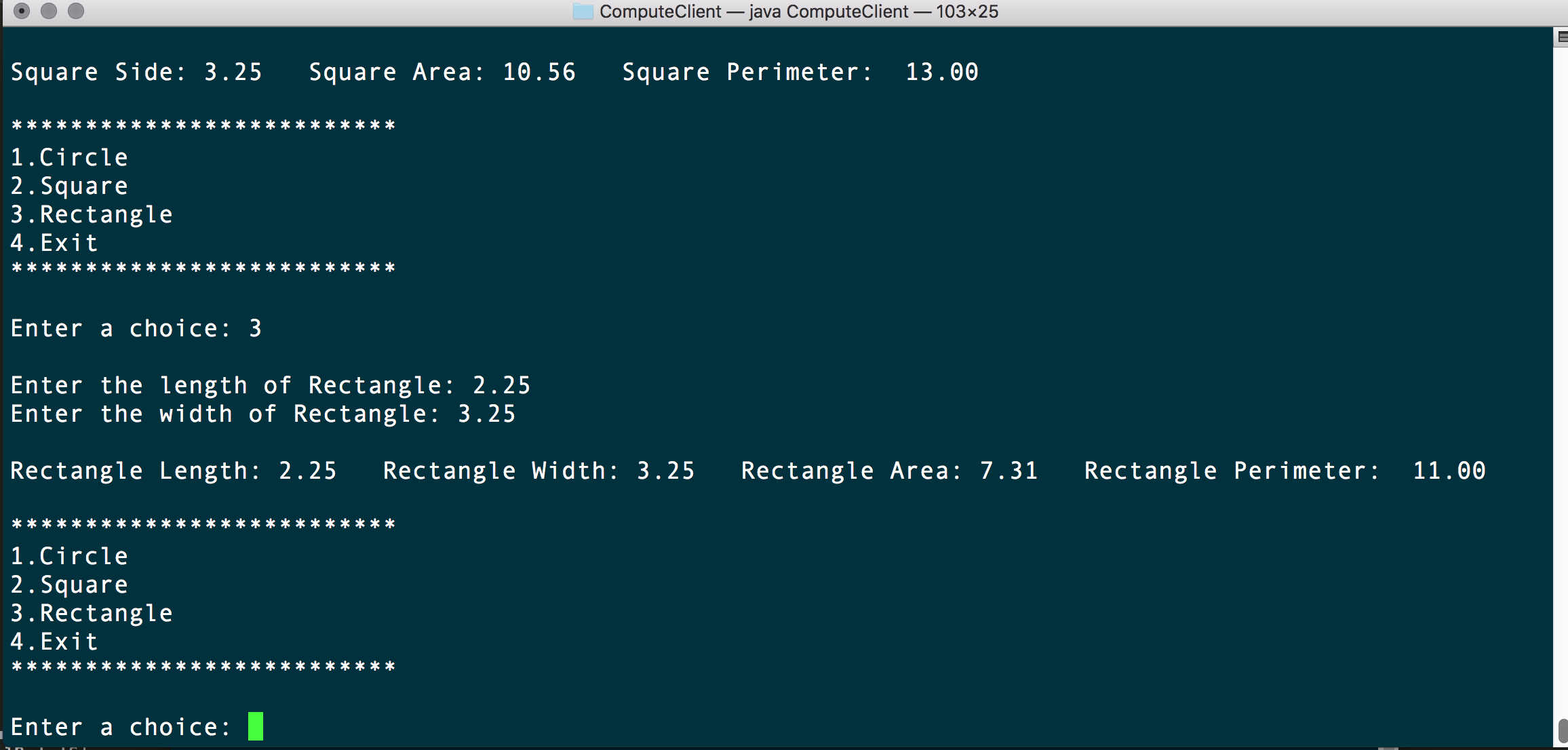
****

**Server Side (ComputeServer)**

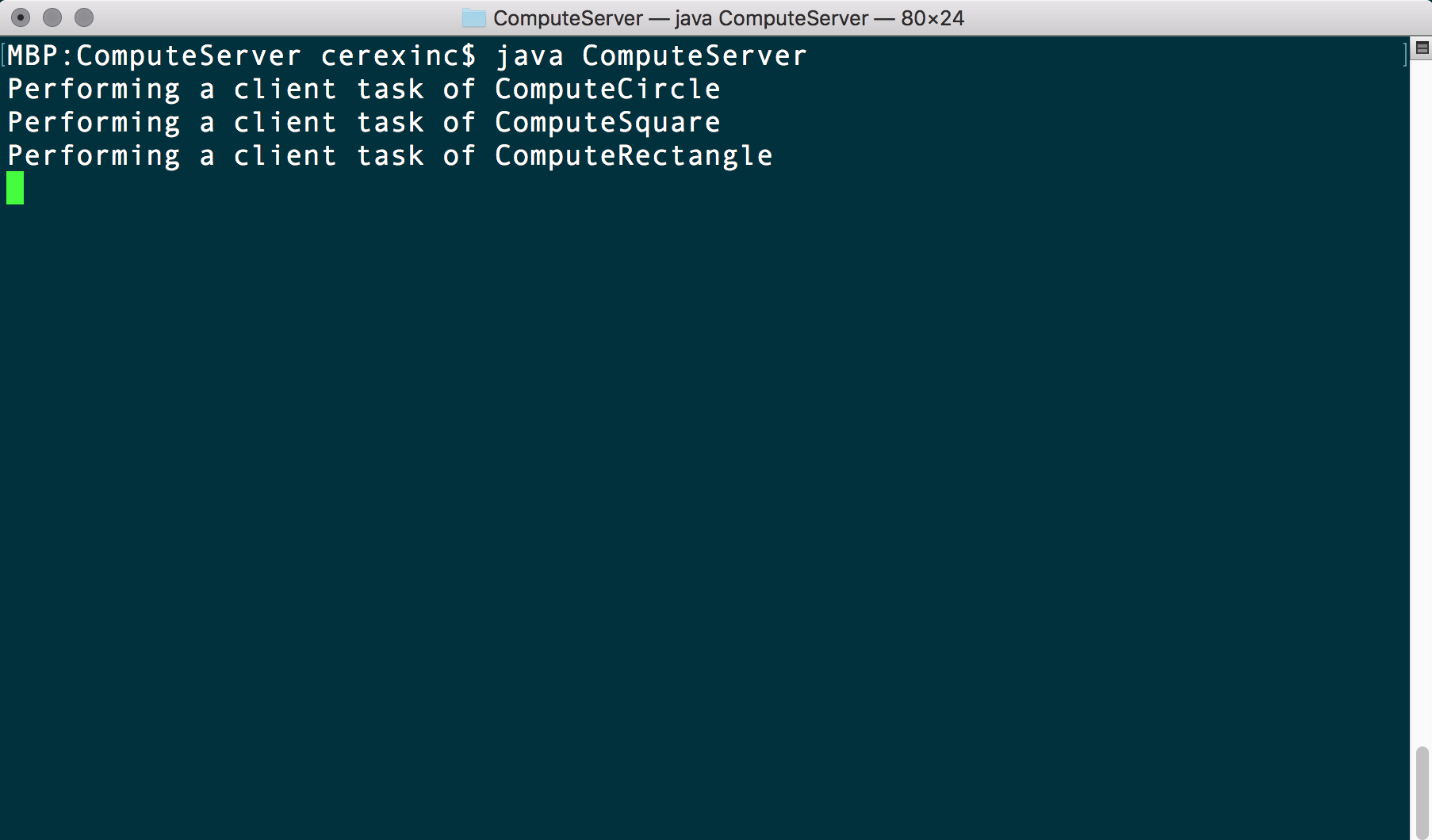
****

**Compute-task Rectangle**

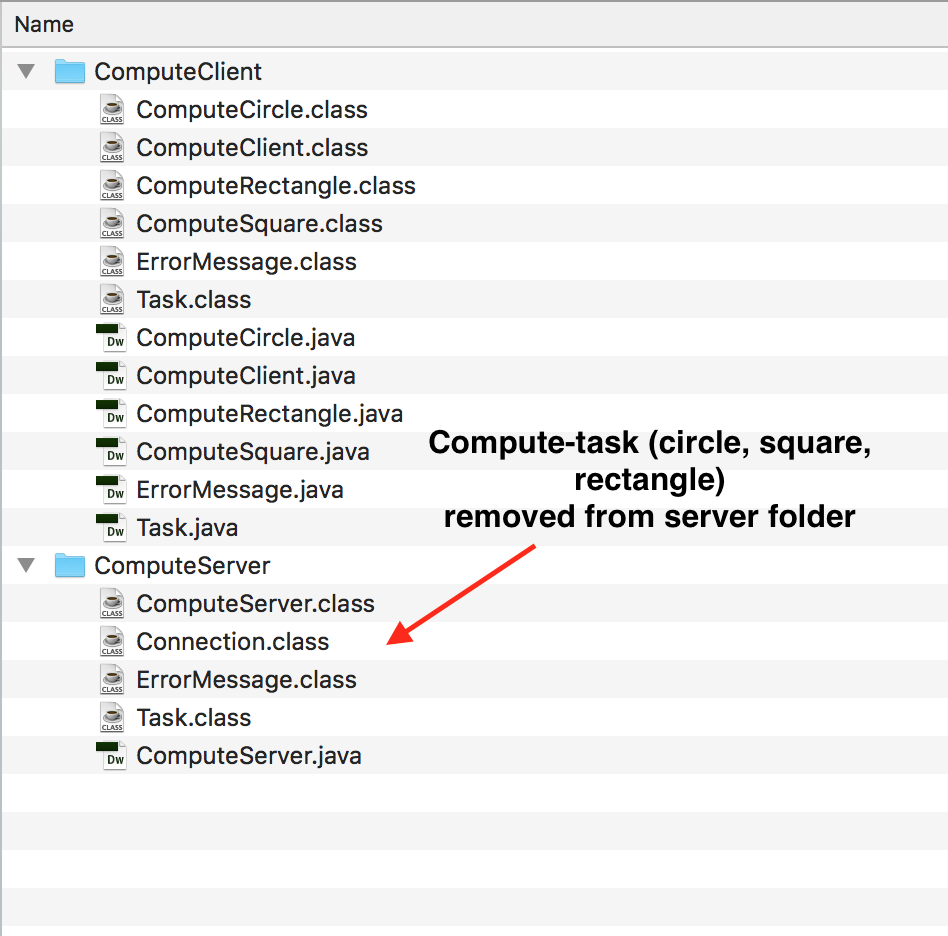
**Client side (ComputeClient)**

****

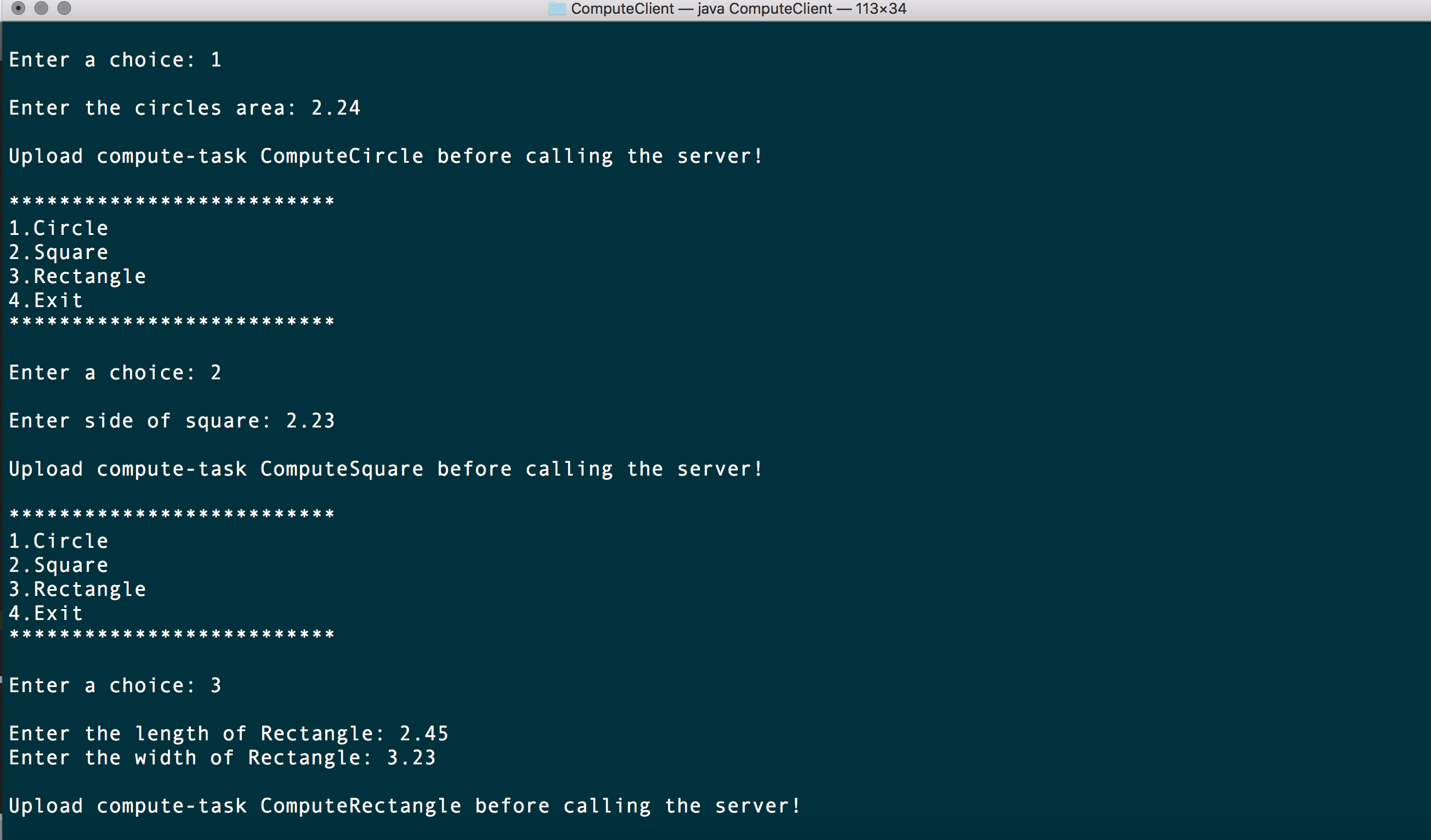
**Server side (ComputeServer)**

****

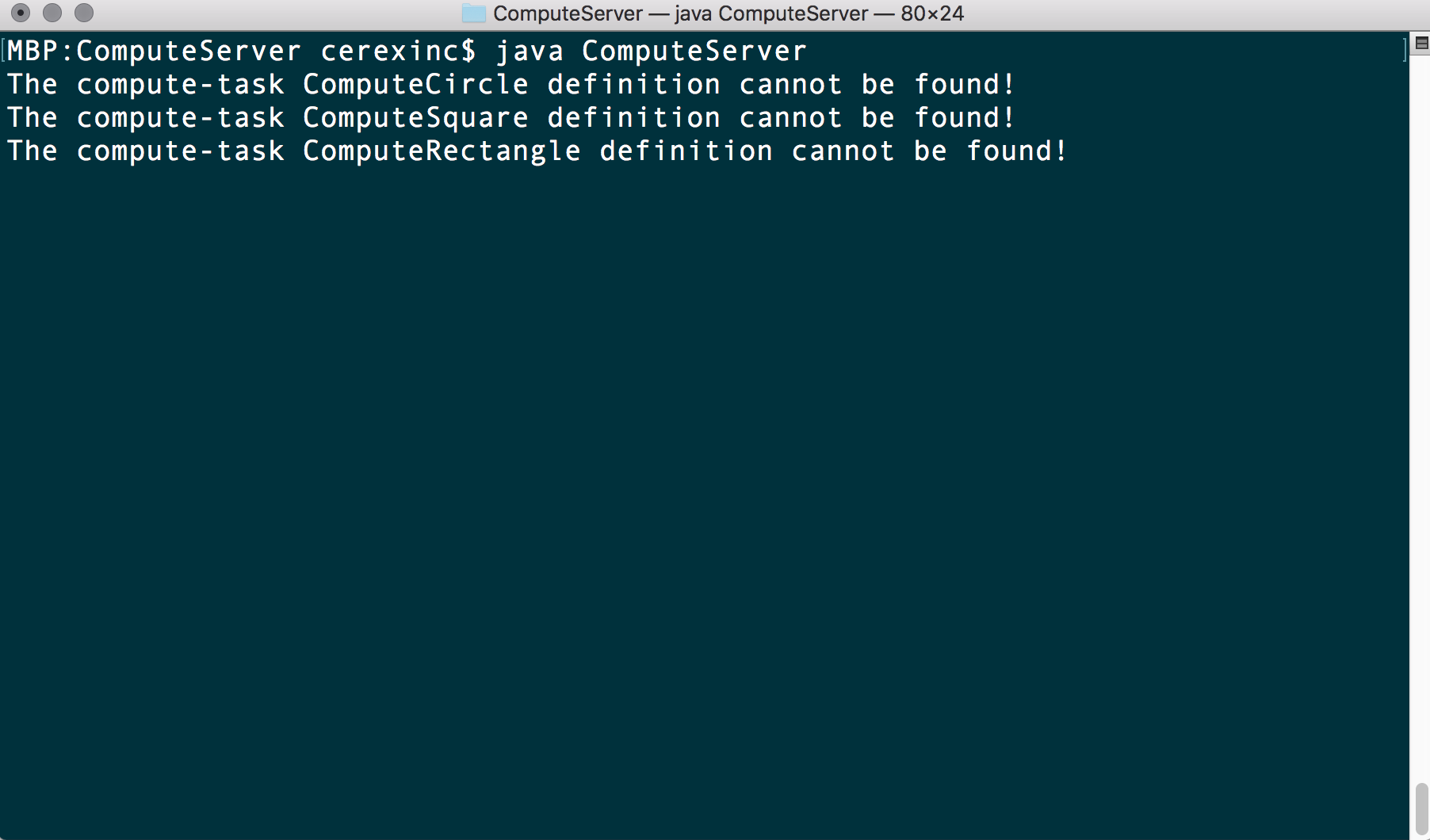
**Exception handling (ClassNotFoundException)**

****

**Client side (ComputeClient)**

****

**Server side (ComputeServer)**

****

**Part 3: Compare and contrast Assignment1 (UDP) with Assignment2 (TCP)**

In assignment one we used the User Datagram Protocol (UDP), an unreliable, connectionless protocol which does not guarantee the delivery of packets and does not order packets or provide any duplication protection (Diffen, 2018).

Overhead in UDP is low due to containing small datagram headers and a lack of network management traffic, making it ideal for applications such as online multiplayer gaming, VOIP videoconferencing, and video streaming services such as Netflix that require fast transmission and can tolerate loss of data (2018). UDP is also ideal for servers who answer small queries from a large number of clients. Protocols that use UDP also include DNS, DHCP, TFTP, SNMP, and RIP (2018).

In assignment two we used the Transmission Control Protocol (TCP) which compared to UDP ensures a reliable and ordered delivery of a stream of bytes from client to server (2018). TCP connections use a three-way handshake which initiates and acknowledges a connection and guarantees delivery by acknowledging packets were received and retransmits data when parts of a message are lost (2018).

TCP provides error checking and also allows ordering by transmitting data in a sequence and reordering data segments when messages are received in the wrong order. TCP is best suited for applications that require high reliability and where transmission time is less critical such as e-mail, file transferring, and web browsing (2018).

TCP has higher overhead compared to UDP due to the three-way handshake process and the acknowledgement, sequencing, and checksum headers applied to transmit data.

**References**

Diffen. 2018. *TCP vs UDP*. [ONLINE] Available at: <https://www.diffen.com/difference/TCP_vs_UDP#TCP_vs._UDP_for_Game_Servers>. [Accessed 18 May 2018].