A **generic communication software** can help two people communicate through voice call, video call, messaging. Before making a voice/video call/message to someone, it sets up a connection printing a message: “**Trying to connect with X**”, where X is the receiver’s name. Before receiving a voice/video call/message to someone, it sets up a connection printing a message: “**X is** **trying to connect with you**”, where X is the sender’s name.

A **messenger** is a type of generic communication software, which can send and receive text, as well as doing all the functionalities of generic communication software. While texting, it prints a message: “**Sending to X the message: Y**”, where X is the receiver’s name and Y is the message sent. While receiving a text, it prints a message: “**X sent you the message: Y**”, where X is the sender’s name and Y is the message sent.

A **modern messenger** is a type of messenger which can send and receive photos, as well as doing all the functionalities of a messenger. While sending a photo, it prints a message: “**Sending to X the image: Y.jpg**”, where X is the receiver’s name and Y is the image sent. While receiving a image, it prints a message: “**X sent you the image: Y.jpg**”, where X is the sender’s name and Y is the message sent.

A **voice caller** is a type of **generic communication software**, which can send and receive voice call, as well as doing all the functionalities of generic communication software. While calling, it prints a message: “**Calling X**”, where X is the receiver’s name. While receiving a call, it prints a message: “**X is calling you**”, where X is the sender’s name.

A **video caller** is a type of **voice caller**, which can send and receive video call, as well as doing all the functionalities of **voice caller**. While video calling, it prints a message: “**Video calling X**”, where X is the receiver’s name. While receiving a call, it prints a message: “**X is video calling you**”, where X is the sender’s name.

A **ModernMessengerAPP** has all attributes of both a modern messenger and a video caller, as well as offer one additional feature. **FacebookMessenger** and **FacebookLiteMessenger are ModernMessengerApps.** **They do everything in the same way. Except two things:** In **FacebookMessenger** you can play game as an additional feature. It prints “**playing X**” then, where **X is the game**. Whereas In **FacebookLiteMessenger** you can play a song as an additional feature. It prints “**playing the song X**” then, where **X is the song**. Moreover, before sending or receiving a photo, **FacebookLiteMessenger** prints a line “**Make sure you have your data mode on”. FacebookMessenger** does not do it. (Think how you can use abstract class/ inheritance here)

**Your Task:**

Implement the whole scenario using inheritance/abstract class/interface. For the methods’ implementation, just use system.out.println. Remember to pass **X,Y as String parameters in methods.**

Your first job is to draw a hierarchy diagram in a page. It will help you to understand the relation between the entities. Remember Khichuri and richfoods diagram? [6 Marks]

Then start from top and develop the classes one by one. [20 Marks]

Also, show a demo in the main function by instantiating an object of **FacebookMessenger & FacebookLiteMessenger** individually and calling their methods. [4 marks]