Pseudocode

dcgps.py

main

```
waitForUserInput
```

gps_utils.py

waitForUserInput()

```
print welcome prompt
while true
     get user input
     if userinput == start
          continuousRead
     else if userinput == exit
          terminate program
```

continousRead()

```
initialize gps socket
initialize gps data stream
connect gps socket to port
set the gps json tag

for incoming data in gps socket
     if incoming data
          store the incoming data in json buffer
          print the gps data
```

gpsprint.py

printData(gpsData)

```
get gps satellites

if satellites > 0
     for satellite in satellites
          get prn
          get elevation
          get azimuth
          get snr
          get used flag
```

```
            get time stamp
            get latitude / longitude
        format output
        print output
```

decimalToDegMinSec(decimalStr)

```
    decimalFloat <- (float) decimalStr.split
    degreesInt <- (int) decimalFloat
    floatPart <- truncateFloat(decimalFloat)
    minutesFloat <- floatPart * 60
    minutesInt <- (int) minutesFloat
    floatPart <- truncateFloat(minutesFloat)
    secondsFloat <- floatPart * 60
    degTuple <- (degreesInt, minutesInt, secondsFloat)
    return degTuple
```

truncateFloat(wholeFloat)

```
    int part <- (int) wholeFloat
    floatPartLen <- len(str(wholeFloat)) – len(str(intPart))
    floatPartWithJunk <- wholeFloat – intPart
    pattern <- . + str(floatPartLen) + .f
    floatPartStr <- format(floatPartWithJunk, pattern)
    return float(floatPartStr)
```