**“Helper.h”**

**(Because of different format code may appear different)**

#ifndef HELPER\_H

#define HELPER\_H

#include <ctime>

#include <string>

#include <sstream>

#include <fstream>

namespace Helper // custom namespace to store our custom data types

{

template <class T>

std::string ToString(const T &);

struct DateTime

{

int D, m, y, M, H, S;

DateTime()

{

time\_t ms;

time(&ms);

struct tm\* info = localtime(&ms);

// format data from info

D = info->tm\_mday;

m = info->tm\_mon + 1; // need to add 1 since january is represented 0

y = 1900 + info->tm\_year; // reference year since C came in the 70s and locatime returns time from that particular day until present day

M = info->tm\_min;

H = info->tm\_hour;

S = info->tm\_sec;

}

DateTime(int D, int m, int y, int M, int H, int S) : D(D), m(m), y(y), M(M), H(H), S(S) {}

DateTime(int D, int m, int y) : D(D), m(m), y(y), M(0), H(0), S(0) {}

DateTime Now() const

{

return DateTime(); // return current date time

}

std::string GetDateString() const

{

// Generate the current date that is correctly formatted in string

return std::string(D < 10 ? "0" : "") + ToString(D) +

std::string(m < 10 ? ".0" : ".") + ToString(m) + "." + ToString(y);

}

std::string GetTimeString(const std::string &sep = ":") const// reference is to default separator which is set to a colon

{

// Generate the current time that is correctly formatted in string

return std::string(H < 10 ? "0" : "") + ToString(H) + sep +

std::string(M < 10 ? "0" : "") + ToString(M) + sep +

std::string(S < 10 ? "0" : "") + ToString(S);

}

std::string GetDateTimeString(const std::string &sep = ":") const

{

return GetDateString() + " " + GetTimeString(sep);

}

};

template <class T>

std::string ToString(const T &e) // only able types that supports the insertion operator

{

std::ostringstream s;

s << e;

return s.str();

}

// OPTIONAL FUNCTION FOR DEBUGGING PURPOSES

void WriteAppLog(const std::string &s) // reference to const string we wish to log

{

std::ofstream file("AppLog.txt", std::ios::app); // app stands for append file

file << "[" << Helper::DateTime().GetDateTimeString() << "]" << "\n" << s << std::endl << "\n";

file.close();

}

}

#endif // HELPER\_H