

# Programming Project

Terry Marris January 2010

## 8 Filenames

We allow the user to set the file names.

### 8.1 filename.h

We have four files, one for entries, one for runners, and two for results. We keep the filenames in a file so that they may be used when required. The user can update the file by changing the filenames stored.

```

/* filename.h: interface */
/* ver 1.0 23Jan2011 */

#ifndef FILENAME_H
#define FILENAME_H

#define FILENAMESIZE 20

typedef struct FILENAME {
    char entry[FILENAMESIZE];
    char runner[FILENAMESIZE];
    char hmresult[FILENAMESIZE];
    char fmresult[FILENAMESIZE];
} FILENAME;

/* fileNameError: displays error message, halts program */
void fileNameError(char *report);

/* newFileName: returns a new file name structure initialised
   with default filenames */
FILENAME newFileName();

/* entryFileName: returns entry filename */
char *entryFileName(FILENAME fileName);

/* runnerFileName: returns runner filename */
char *runnerFileName(FILENAME fileName);

/* halfMarathonFileName: returns half marathon results filename */
char *halfMarathonFileName(FILENAME fileName);

/* fullMarathonFileName: returns runner filename */
char *fullMarathonFileName(FILENAME fileName);

/* setEntryFileName: amends entry file name */
FILENAME setEntryFileName(char *entryFileName, FILENAME fileName);

/* setRunnerFileName: amends runner file name */
FILENAME setRunnerFileName(char *runnerFileName, FILENAME fileName);

```

```

/* setHalfMarathonFileName: amends half marathon file name */
FILENAME setHalfMarathonFileName(
    char *hmFileName, FILENAME fileName);

/* setFullMarathonFileName: amends full marathon file name */
FILENAME setFullMarathonFileName(
    char *fmFileName, FILENAME fileName);

/* getEntryFileName: returns entry file name from the keyboard */
char *getEntryFileName();

/* getRunnerFileName: returns runner file name from the keyboard */
char *getRunnerFileName();

/* getHalfMarathonFileName: returns half marathon file name from the
   keyboard */
char *getHalfMarathonFileName();

/* getFullMarathonFileName: returns full marathon file name from the
   keyboard */
char *getFullMarathonFileName();

/* getFileName: gets entry, runner and results file names from
   the keyboard */
FILENAME getFileName(FILENAME fileName);

/* printFileName: displays file names */
int printFileName(FILENAME name);

/* writeToFile: writes file names to a file */
int writeToFile(FILENAME name);

/* readFromFile: retrieves file names from file */
FILENAME readFromFile();

#endif

```

## 8.2 filename.c

```

/* filename.c: maintains a file of file names */
/* ver 1.0 23Jan2011 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "utility.h"
#include "filename.h"

#define xFileName "filename.dat"

/* fileNameError: displays error message, halts program */
void fileNameError(char *report)
{
    printf("\n%s\n", report);
    exit(EXIT_FAILURE);
}

```

```

/* newFileName: returns a new file name structure initialised
   with default filenames */
FILENAME newFileName()
{
    FILENAME fileName;
    strcpy(fileName.entry, "entries2010.dat");
    strcpy(fileName.runner, "run2010.dat");
    strcpy(fileName.hmresult, "hmresults2010.dat");
    strcpy(fileName.fmresult, "fmresults2010.dat");
    return fileName;
}

/* entryFileName: returns entry filename */
char *entryFileName(FILENAME fileName)
{
    char *name = malloc(FILENAMESIZE);
    strcpy(name, fileName.entry);
    return name;
}

/* runnerFileName: returns runner filename */
char *runnerFileName(FILENAME fileName)
{
    char *name = malloc(FILENAMESIZE);
    strcpy(name, fileName.runner);
    return name;
}

/* halfMarathonFileName: returns half marathon results filename */
char *halfMarathonFileName(FILENAME fileName)
{
    char *name = malloc(FILENAMESIZE);
    strcpy(name, fileName.hmresult);
    return name;
}

/* fullMarathonFileName: returns runner filename */
char *fullMarathonFileName(FILENAME fileName)
{
    char *name = malloc(FILENAMESIZE);
    strcpy(name, fileName.fmresult);
    return name;
}

/* setEntryFileName: amends entry file name */
FILENAME setEntryFileName(char *entryFileName, FILENAME fileName)
{
    FILENAME name = fileName;
    strcpy(name.entry, entryFileName);
    return name;
}

/* setRunnerFileName: amends runner file name */
FILENAME setRunnerFileName(char *runnerFileName, FILENAME fileName)
{
    FILENAME name = fileName;
    strcpy(name.runner, runnerFileName);
    return name;
}

```

```

/* setHalfMarathonFileName: amends half marathon file name */
FILENAME setHalfMarathonFileName(char *hmFileName, FILENAME fileName)
{
    FILENAME name = fileName;
    strcpy(name.hmresult, hmFileName);
    return name;
}

/* setFullMarathonFileName: amends full marathon file name */
FILENAME setFullMarathonFileName(char *fmFileName, FILENAME fileName)
{
    FILENAME name = fileName;
    strcpy(name.fmresult, fmFileName);
    return name;
}

/* getEntryFileName: returns entry file name from the keyboard */
char *getEntryFileName()
{
    char *name = malloc(FILENAMESIZE);
    if (name == NULL)
        fileNameError("getEntryFileName: out of memory");
    return getStr("Entry file name?", FILENAMESIZE);
}

/* getRunnerFileName: returns runner file name from the keyboard */
char *getRunnerFileName()
{
    char *name = malloc(FILENAMESIZE);
    if (name == NULL)
        fileNameError("getEntryFileName: out of memory");
    return getStr("Runner file name?", FILENAMESIZE);
}

/* getHalfMarathonFileName: returns half marathon file name from the
keyboard */
char *getHalfMarathonFileName()
{
    char *name = malloc(FILENAMESIZE);
    if (name == NULL)
        fileNameError("getEntryFileName: out of memory");
    return getStr("Half marathon file name?", FILENAMESIZE);
}

/* getFullMarathonFileName: returns full marathon file name from the
keyboard */
char *getFullMarathonFileName()
{
    char *name = malloc(FILENAMESIZE);
    if (name == NULL)
        fileNameError("getEntryFileName: out of memory");
    return getStr("Full marathon file name?", FILENAMESIZE);
}

```

```

/* getFileName: gets entry, runner and results file names from
the keyboard */
FILENAME getFileName(FILENAME fileName)
{
    FILENAME name;
    name = setEntryFileName(getEntryFileName(), name);
    name = setRunnerFileName(getRunnerFileName(), name);
    name = setHalfMarathonFileName(getHalfMarathonFileName(), name);
    name = setFullMarathonFileName(getFullMarathonFileName(), name);
    return name;
}

/* printFileName: displays file names */
int printFileName(FILENAME name)
{
    printf("%s, %s, %s, %s\n", entryFileName(name),
        runnerFileName(name), halfMarathonFileName(name),
        fullMarathonFileName(name));
    return 0;
}

/* writeToFile: writes file names to a file */
int writeToFile(FILENAME name)
{
    FILE *file = fopen(xFileName, "wb");
    if (file == NULL)
        fileNameError("writeToFile: cannot open file");
    fwrite(&name, sizeof(FILENAME), 1, file);
    fclose(file);
    return 0;
}

/* fileExists: returns 1 (true) if file with given name exists */
int fileExists()
{
    if (fopen(xFileName, "rb") == NULL)
        return 0;
    return 1;
}

/* readFromFile: retrieves file names from file */
FILENAME readFromFile()
{
    FILENAME name;
    FILE *file;

    if (!fileExists()) {
        name = newFileName();
        writeToFile(name);
    }
    file = fopen(xFileName, "rb");
    if (file == NULL)
        fileNameError("readFromFile: cannot open file");
    fread(&name, sizeof(FILENAME), 1, file);
    while (!feof(file))
        fread(&name, sizeof(FILENAME), 1, file);
    fclose(file);
    return name;
}

```

*filename.c* is tested in *marathonmenu*.