EE101 C programming and SW engineering 1 Example Application of Software Development Process

The following gives you an example on how to complete a software project by following the Software Development Process (SDP) step by step. This is also what you should follow to write up your assignment reports.

Example: (for each assignment, you will be given an assignment sheet similar to this) Write a C program which takes a depth into the earth, given in units of kilometers, as the input data and compute and display the corresponding temperature at this particular depth in two formats, namely, degrees Celsius and degrees Fahrenheit. The relevant formulas are given as follows:

- Temperature in degrees Celsius given the depth in kilometers:

  Celsius = 10 * depth + 20
- Temperature in degrees Fahrenheit is obtained by converting temperature in degrees Celsius:

  Fahrenheit = 9/5 * Celsius + 32

Software Development Process

1) Problem statement:

In this part, your job is to fully understand the requirements of this project, as proposed by the user of this software (your customer) so that all the functionalities this software needs to provide are clearly understood and stated. Given the above example, you may list all the functionalities this program should offer as follows:

1) Provide an interface through which the user can input a number which is a real number and represents a legal depth in kilometers.
2) The execution of the program will be terminated if an illegal input is detected. Being illegal may be defined as follows:
   - The input depth is out of the legal range (if we know the legal range and, here, it is left out of our considerations for simplicity).
   - The input contains non-numerical characters.
3) The program should display two outputs: degrees Celsius and degrees Fahrenheit. These two temperatures should have an accuracy of 2 digits after the decimal point (Note: the original statement did not specify this, common sense is applied by the programmer and agreed with the customer following production of the problem statement).
4) The program exits.

2) Analysis

In this part, your job is to analyze all the requirements listed in above section to decide the entire software architecture and chose the best data structure as well as the most suitable algorithm.

For the example at hand, it is clear that we need to deal with the Input/Output and turn the mathematical equations into C codes.

- **On an input:**
  Firstly, a message needs to be printed on the screen to let the users know what data/action this program expects. Because the execution is terminated upon the detection of an illegal input, we can specify the desired data type in the c code used to read the input off the keyboard.

- **On Outputs:**
  Given a legal input, two temperatures can be easily calculated and printed on the screen along with an explanatory message.

- **Data structure:**
  Only single data is involved. So there are no complex data structure issues here.

- **Algorithm:**
  Evaluating the above two equations is straight forward. So there are no algorithm issues here either.

3) Design

In this part, you need to give a relatively detailed procedure, on which the implementation of the code can be based. In this example, a very detailed design is given just because this example is extremely simple. In the future, as the complexity of the assignments goes up significantly, you can give an overview on your entire software architecture just by using a flow chart or other means instead of giving a detailed design description like this.

1. Declare three variables of type “float” and name them appropriately e.g. depth, celsius and fahrenheit, respectively:
   depth – to store and represent the input depth in kilometers.
   celsius - to store and represent the temperature in degrees Celsius.
   fahrenheit - to store and represent the temperature in degrees Fahrenheit.
2. Print a message on the screen to ask the user to input a real number representing the

depth in kilometers.

3. Read the input real number from the keyboard and assign this value to the variable named depth.
4. The program is terminated if the read fails. (The data type of the input data does not meet the data type specified)
5. Compute the temperature in degrees Celsius, if the reading is successful, according to the provided formula and store the value in the variable named “celsius”.
6. Compute the temperature in degrees Fahrenheit according to the provided formula and store the value in the variable named “fahrenheit”.
7. Display the two temperatures on the screen with an accuracy of 2 digits after the decimal point.

4) Implementation

see the C code “sample.c” (at the end of this document) with comments.

5) Testing

The C program can be tested by carrying out a set of data input tests using various valid inputs first and then using various invalid inputs (important). Given the corresponding inputs, the C program behaves exactly as expected. Some test cases are shown below:

Test 1

```
Please enter the depth in kilometers inside the earth as input data: 10
Celsius temperature at depth 10.000000 in km is 120.00.
Fahrenheit temperature at depth 10.000000 in km is 248.00
```

Test 2

```
Please enter the depth in kilometers inside the earth as input data: 250.5
Celsius temperature at depth 250.500000 in km is 2525.00.
Fahrenheit temperature at depth 250.000000 in km is 4577.00.
```

Test 3

```
Please enter the depth in kilometers inside the earth as input data: 5000
Celsius temperature at depth 5000.000000 in km is 50020.00.
Fahrenheit temperature at depth 5000.000000 in km is 90068.00.
```

Test 4

```
Please enter the depth in kilometers inside the earth as input data: a
Program terminates
```

Test 5

```
Please enter the depth in kilometers inside the earth as input data: Program
Program terminates
```

Note: Testing should be more extensive than described above to give a much wider coverage on different scenarios. This means more illegal inputs should be attempted and a note should be taken on cases that the program fails to produce the expected outputs. For example, in the case that the value entered through the keyboard begins with a digit followed by a non-numerical character (e.g. 1,1 or 1n1), what happens is that only the first numerical digit was read and, therefore, the outputs were calculated based on that digit only. In such a case, you should define the sensible desired behavior by yourself. You should try your best to find the bugs your program fails to handle.

C code

```
/*
Name: Simple Program for Depth, Degrees Celsius and Degree Fahrenheit Conversion
File Name: sample.c
Copyright: Free

Author: Anonymous Author

Description: Compute and display the temperature at a given depth inside the earth in
degrees Celsius and degrees Fahrenheit
*/

#include <stdio.h>     /* include standard library stdio.h for use of scanf() and
                       printf() */
#include <stdlib.h> /* include standard library stdlib.h for use of exit() */

int main(void){                  /* define a function named main */

float depth, celsius, fahrenheit;         /* declare three variables of type of float namely
                                          depth, Celsius and Fahrenheit */

printf("Please enter the depth in kilometers inside the earth as input data:\n");
/* ask user to input a real number representing the variable depth in kilometers */

if(scanf("%f", &depth)!=1)   /* store the value of the input real number into
                             variable depth using the scanf function */
          exit(0);           /* test the return value of scanf to make sure the read
                             was successful. If the read failed exit the program */

celsius = 10*depth + 20;     /* compute the temperature at the given depth in
                             degrees Celsius according to the formula provided */

fahrenheit = 1.8*celsius + 32;            /* compute the temperature at the given depth
                                          in degrees Fahrenheit according to the formula
                                          provided */

printf("Celsius temperature at depth %.2f in km is %.2f.\n", depth, celsius);
/* display the Celsius temperature at the given depth */

printf("Fahrenheit temperature at depth %.2f in km is %.2f.\n", depth, fahrenheit);
/* display the Fahrenheit temperature at the given depth */

return 0;
}
```

Notes

The solution of this exercise demonstrates solving practical problems in C using the Software Development Process. Students are expected to present their work using a similar approach as presented above.