Write a simple program in C

Session 1 15 October 2002
Achieve Target 1

The following program writes $\text{Height} = 1.700000$ to the screen.

```c
#include <stdio.h>

void main() {
    float height = 1.70;
    printf("Height = %f", height);
    getchar();
}
```

The four lines in blue appear in almost all C programs. **Remember** them.
Achieve Target 1

We now want the program to write Weight data as well.

```c
#include <stdio.h>

void main() {
    float height = 1.70;
    printf("Height = %f", height);
    getchar();
}
```

Work on the program to make it happen.
Achieve Target 1

Solution

```c
#include <stdio.h>

void main() {
    float height = 1.70;
    float weight = 70;
    printf("Height = %.2f", height);
    printf("Weight = %.2f", weight);
    getchar();
}
```

The output of the revised program may not be nice.
Discuss the problem and how to make it look clearer. The output is not nice because the height and weight should be printed on separate line. The trick is to add the escape sequence "\n" to move the cursor to the next line at the right place.

```c
#include <stdio.h>

void main() {
    float height = 1.70;
    float weight = 70;
    printf("Height = %f\n", height);
    printf("Weight = %f", weight);
    getchar();
}
```
The following program makes a conversion of weight from kilogram to pound.

```c
#include <stdio.h>

void main() {
    float weight = 70; /* in kilogram */
    float pound;

    pound = 2.2 * weight;

    printf("%f kg = %f pound", weight, pound);

    getchar();
}
```

The line in red makes the conversion calculation from kilogram to pound.
Discuss how it happens.

```c
#include <stdio.h>
void main() {
    float weight = 70; /* in kilogram */
    float pound;
    pound = 2.2 * weight;
    printf("%f kg = %f pound", weight, pound);
    getchar();
}
```

The line in red makes the conversion calculation from kilogram to pound.
Achieve Target 2

We now want the program to calculate the Body Mass Index (BMI).

BMI is calculated using weight and height of a person by the following formula. **Work** on the program to make it happen.

\[
BMI = \frac{\text{weight}(\text{kg})}{\text{height}(\text{m}) \times \text{height}(\text{m})}
\]

```c
#include <stdio.h>

void main() {
    float weight = 70; /* in kilogram */
    float pound;
    pound = 2.2 * weight;
    printf("%.f kg = %.f pound", weight, pound);
    getchar();
}
```
#include<stdio.h>

/* BMI = weight(kg) / (height(m) * height(m)) */

void main()
{
    float height = 1.70;
    float weight = 70;

    float bmi;
    bmi = weight / (height * height);
    printf("Height = %f m\n", height);
    printf("Weight = %f kg\n", weight);
    printf("BMI = %f", bmi);
    getchar();
}
The following program again makes a conversion of weight from kilogram to pound, and the weight data is now allow user to enter from keyboard. In other words, the program supports input.

```c
#include <stdio.h>

void main() {
    float weight, pound;
    printf("Enter Weight in kg: ");
    scanf("%f", &weight);
    pound = 2.2 * weight;
    printf("\n%f kg = %f pound", weight, pound);
    fflush(stdin);
    getchar();
}
```
Achieve Target 3

The lines in red control the input of weight. **Discuss** how it happens.

```c
#include <stdio.h>
void main() {
    float weight, pound;  
    printf("Enter Weight in kg: ");
    scanf("%f", &weight);
    pound = 2.2 * weight;
    printf("\n%f kg = %f pound", weight, pound);
    fflush(stdin);
    getchar();
}
```

Note that the variable declaration line (in blue) now has two variables declared in one statement. **Learn** that it is a short form.

The lines in green clear the input in the buffer. **Discuss** what’s happen if we don’t have this line.
Achieve Target 3

Use the BMI program you have written for the previous Target, **work** on it so that it allows the weight and height to be entered.

```c
#include <stdio.h>

void main() {
    float weight, pound;
    printf("Enter Weight in kg: ");
    scanf("%f", &weight);
    pound = 2.2 * weight;
    printf("\n%f kg = %f pound", weight, pound);
    fflush(stdin);
    getchar();
}
```
Achieve Target 3

Solution

```c
#include<stdio.h>
void main()
{
    float height, weight, bmi;

    printf("Enter your Height(m): ");
    scanf("%f", &height);

    printf("Enter your Weight(kg): ");
    scanf("%f", &weight);

    bmi = weight / (height * height);

    printf("Your BMI = %.2f", bmi);
    fflush(stdin);
    getchar();
}
```
Achieve Target 4

This is our final program of this Project Perform Session. This program calculates BMI from the input height and weight.

```c
#include<stdio.h>

void main(){
    float height, weight, bmi;

    printf("Enter your Height(m): ");
    scanf("%f", &height);

    printf("Enter your Weight(kg): ");
    scanf("%f", &weight);

    bmi = weight / (height * height);

    printf("Your BMI = %f", bmi);
    fflush(stdin);
    getchar();
}
```

Study the program and identify the following two features.
1. The lines that nearly always appear in C programs (refer to Target 1).
2. The four basic programming elements – input, output, variables (memory), and operators.
```c
#include<stdio.h>
void main(){
    float height, weight, bmi;
    printf("Enter your Height(m): ");
    scanf("%f", &height);
    printf("Enter your Weight(kg): ");
    scanf("%f", &weight);
    bmi = weight / (height * height);
    printf("Your BMI = %f", bmi);
    fflush(stdin);
    getchar();
}
```

After you have understood the program above, memorize the program.
Achieve Target 4

On a white sheet of paper, **write** the BMI program again without looking at any reference material. Check whether your program has the two features mentioned above.
In the last 4 targets, you have touched on the following ideas.

Ponder upon these ideas and remember what you have learned.

The ideas are arranged so that the next one is more challenging. We expect you at least reach to Idea 3.
Achieve Target 5

1. The basic structure of a C program – what are the lines that nearly appear in every C program?

2. The four common elements of C programs – input, output, variables (memory) and operators. These four appears in the program (from Target 4) that you have memorized. Could you identify them?

3. C programs have a syntax that you need to follow. Every semicolon counts! Beware of the following characters especially and place them at the right place. ( ) { } " ; &

4. Useful C programs must have logic written into it. For example, the kilogram-to-pound conversion program has the logic to do the conversion. The BMI program has the logic to calculate the BMI.

5. The logic allows C programs to process data and turn it into information. For example, the BMI program turns the data of weight and height into BMI, a piece of more useful information. After you have successfully written a C program, identify the data required and information produced by your program.
End of Session 1

You have done well! Keep going!

Session 2 29 October 2002