

IS12 - Introduction to Programming

Lecture 10: Simple Programs

Peter Brusilovsky

<http://www2.sis.pitt.edu/~peterb/0012-051/>

Exchange Kiosk: Assignment

```
void main()
{
    float dollars_for_mark; /* exchange rate */
    int commission; /* commission in dollars */
    float marks; /* marks given */
    float dollars; /* dollars returned */

    /* get data */
    dollars_for_mark = 0.666;
    commission = 3;
    marks = 100;

    /* calculate USD */
    dollars = marks * dollars_for_mark - commission;

    /* print result */
    printf("For %.2f marks you will get %.2f dollars!\n" , marks,
dollars);
}
```



Exchange Kiosk: Initialization

```
void main()
{
    float dollars_for_mark = 0.666; /* exchange rate */
    int commission = 3; /* commission in dollars */
    float marks = 100; /* marks given */
    float dollars; /* dollars returned */

    /* calculate USD */
    dollars = marks * dollars_for_mark - commission;

    /* print result */
    printf("For %6.2f marks you will get %6.2f dollars!\n" , marks,
dollars);
}
```



Exchange Kiosk: #define

```
#define DOLLARS_FOR_MARK 0.666
#define COMMISSION 3
#define MARKS 100

void main()
{
    float dollars; /* dollars returned */

    /* calculate USD */
    dollars = MARKS * DOLLARS_FOR_MARK - COMMISSION ;

    /* print result */
    printf("For %6.2f marks you will get %6.2f dollars!\n" , MARKS,
dollars);
}
```



Defined constants

- `#define` is a preprocessor directive for defining constants
- Example:
`#define COMMISSION 3`
`COMMISSION` is a *defined constant*
 - Now every appearance of `commission` is literally replaced by `3`
 - It is not a variable, it can't be assigned a value



Why to use defined constants?

- Remember example 2.1?

```
/* Temperature Converter */
#include <stdio.h>

void main()
{
    printf("100 Fahrenheit = %.2f Celsius\n",
        (5.0/9.0)*(100-32));
}
```
- Need to replace 100 to 150 in two places (in real programs in dozens places!)

Example: Temperature Converter

```
/*
   Example: Temperature Converter
   Author: Peter Brusilovsky
   Objective: use of #define and printf
*/
#include <stdio.h>
#define FAHR 100.0 /* the temperature to be converted */

void main()
{
    printf ("%6.2f Fahrenheit = %6.2f Celsius\n",
           FAHR, (5.0 / 9.0) * (FAHR - 32));
}
```

More About Preprocessor

- Preprocessor is a separate part of C compiler, which differs from the rest of it
- Makes file insertions
 - #include <stdio.h> - system files
 - #include "myfile.h" - user files
- Resolves definitions
 - #define NUM 100.0
- Removes comments
 - /* this is simply a comment */



Example after preprocessing

...content of the file stdio.h...

```
void main()  
{  
    printf("%6.2f Fahrenheit = %6.2f Celsius\n",  
          100.0, (5.0 / 9.0) * (100.0 - 32));  
}
```



Constants are not variables!

```
#include <stdio.h>  
#define FAHR 100.0 /* the temperature to be converted */  
  
void main()  
{  
    printf("%6.2f Fahrenheit = %6.2f Celsius\n",  
          FAHR, (5.0 / 9.0) * (FAHR - 32));  
    FAHR = 80;  
    printf("%6.2f Fahrenheit = %6.2f Celsius\n",  
          FAHR, (5.0 / 9.0) * (FAHR - 32));  
}
```

... after preprocessing

...content of the file stdio.h...

```
void main()  
{  
    printf("%6.2f Fahrenheit = %6.2f Celsius\n",  
          100.0, (5.0 / 9.0) * (100.0 - 32));  
    100.0 = 80;  
    printf("%6.2f Fahrenheit = %6.2f Celsius\n",  
          100.0, (5.0 / 9.0) * (100.0 - 32));  
}
```

Usage Rules

- **Defined *constant*:** `#define COMMISSION 3`
 - This value stays the same in the program.
- **Initialized variable:** `int commission = 3;`
 - The value can be changed in the program, but the initial value is known before the start of the program
- **Assignment:** `commission = 3;`
 - Initial value is not known before the start of the program



Example: Temperature Converter

```
/* Example: Temperature Converter with input
   Author: Peter Brusilovsky
*/
#include <stdio.h>

void main() {
    float fahr; /* the temperature to be converted */

    /* Data input */
    printf("Enter temperature in Fahrenheit: "); /* prompt */
    scanf("%f",&fahr); /* input */

    /* Calculation and printing */
    printf("%6.2f Fahrenheit = %6.2f Celsius\n",
           fahr, (5.0 / 9.0) * (fahr - 32));
}
```



Getting Changeable Data Inside

- Defined constant
- Initialization of variables
- Assignment
- Input data

```
scanf ("%d" ,&marks) ;
```

- The computer will read the symbols from **standard input**, consider it as an integer (%d tells about it) and assign to the given variable marks.
- Treat **&** as a magic sign for now

Example: Better Exchange Kiosk

```
#include <stdio.h>
#define DOLLARS_FOR_MARK 0.666 /* exchange rate */
#define COMMISSION 3 /* commission in dollars */
void main()
{
    int marks;
    float dollars;

    /* get data */
    printf("Marks to exchange?: "); /* prompt */
    scanf("%d",&marks); /* input */

    /* calculate USD */
    dollars = marks * DOLLARS_FOR_MARK - COMMISSION;
    /* print results */
    printf("For %d marks you will get %.2f dollars!\n" , marks ,
dollars);
}
```

Example: Capital growth

```
#include <stdio.h>

void main()
{
    float interest_rate; /* interest rate in percents */
    int capital; /* starting capital in dollars */
    float annual_interest; /* annual interest in dollars */

    printf("Capital ($$) and interest rate (%%xx.x): ");
    scanf("%d %f",&capital, &interest_rate);

    annual_interest = capital * interest_rate / 100;
    printf("Interest %.2f; Total %9.2f\n", annual_interest,
capital + annual_interest);
}
```



More about input

- Input specification is important!

```
scanf("%f",&dollars_for_mark);  
scanf("%d",&marks);
```

- The computer will read the symbols from **standard input**, consider it according the **given input specification** (i.e., %f or %d) and assign to the given variable

- Several values could be entered

```
scanf("%d %f",&capital, &interest_rate);
```



A simple C calculation program

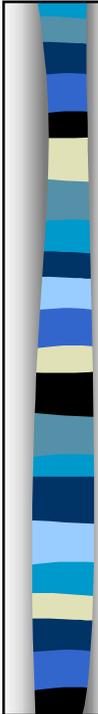
- Analyze the problem

- What we need to calculate
- What data do we need to calculate

- Define variables and constants

- Declare target values as variables
- Define known data as initialized variables or symbolic constants
- Declare unknown data as variables and request the values from the user via input

- Calculate and print result



Before next lecture:

- Do reading assignment (quiz!)
 - Perry: Chapter 7; Chapter 8
- Run Classroom Examples
- Check yourself by working with KnowledgeTree and WADEIn system
- Exercises: Do Celsius to Fahrenheit conversion *with input*
- Suggest a simple calculation problem and write a program with input to solve it