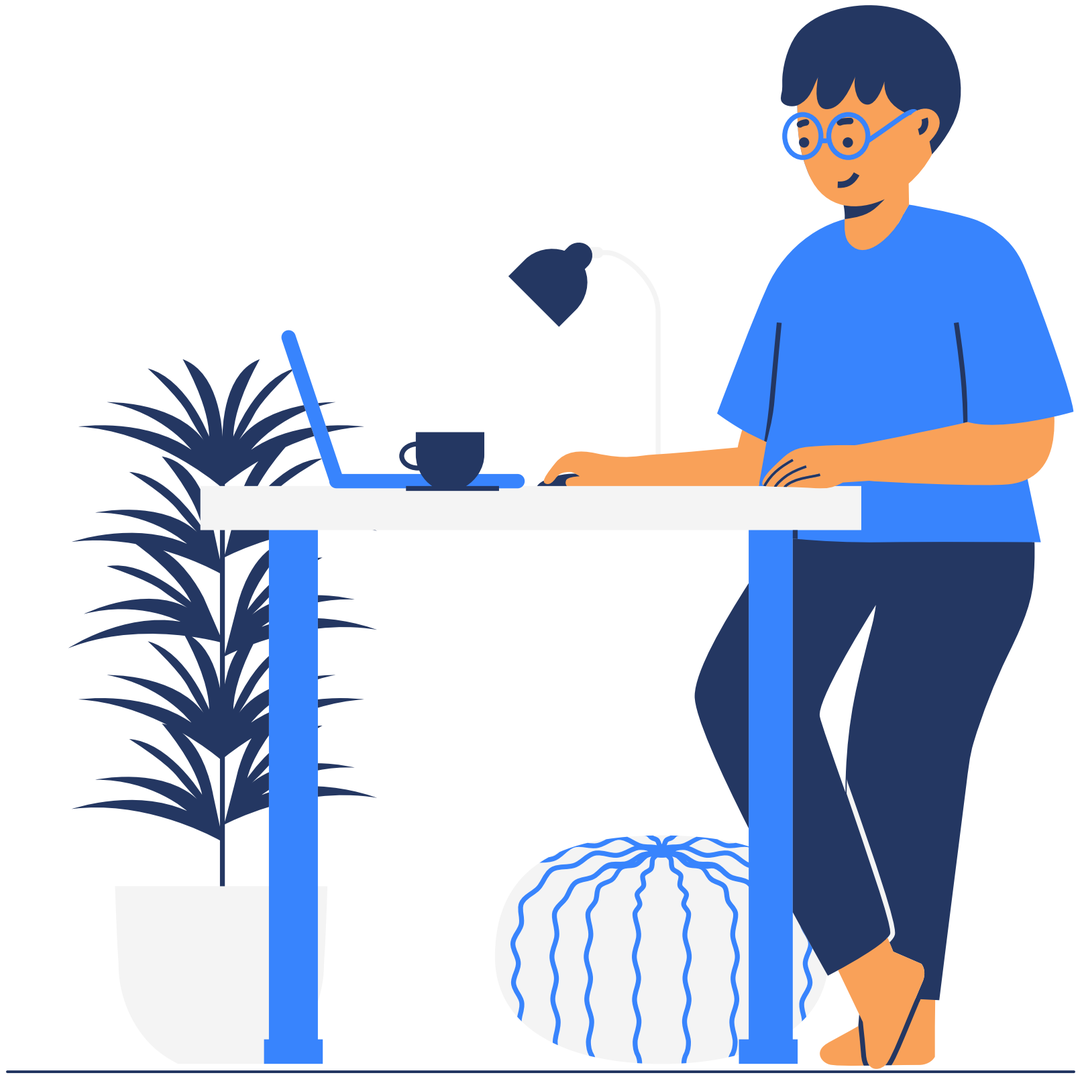
BINUS UNIVERSITY

Input & Output

C Programming

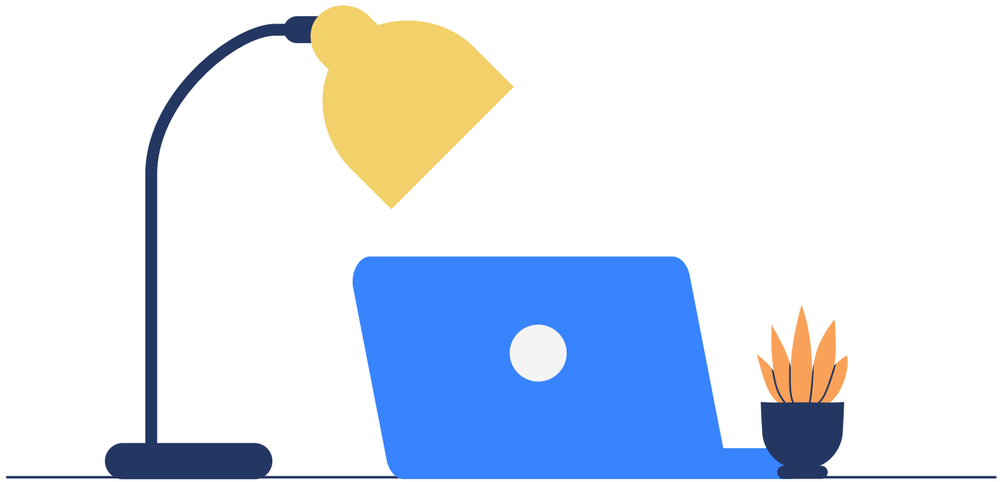


Hi

I’m Anang Prasetyo, a Lecturer from School of Computer Science in Binus University.

Review

Previous session you already learn about varibale, constant, and data type



Variable

Variables are containers for storing data values.

Constant

When you don't want others (or yourself) to override existing variable values, use the const keyword (this will declare the variable as "constant", which means unchangeable and read-only):

Data Type

A variable in C must be a specified data type, and you must use a format specifier inside the printf() function to display it

Functions

We need a function in C to create a some instructions.

Standard library function

Are predefined functions that C compiler provides

Programmer-defined function

User defined function are functions that are created by the programmer/developer to be used inside a program

Output operation

To show data on the display screen/monitor. Some of standard library function in C

- printf();
- putchar();
- putch();
- puts();
- etc.

Output Operations: printf function

To display some data on the standard output, using certain format

- Standard output is the monitor.
- Syntax:

printf(const char *format[,argument, ...]);

- Header file : stdio.h

Output Operations: printf function

```
/* A first program in C */
#include <stdio.h>
void main()
{
      printf (“Welcome to C!\n”);
}
```

```
/*Printing on one line with two printf statements*/
#include <stdio.h>
int main(void){
      printf (“Welcome”);
      printf (“to C!\n”);
      return 0;
}
```

Ouput Formatting

Output also has formatted specification:

%[flags][width][.precision] type

width : number of columns provided

precision : digit number

flags :

can be changed into:

- none : right justify
- - : left justify
- -+ : for positive & negative value

type :

d –or- i :signed decimal

o : unsigned octal

u : unsigned decimal

x : unsigned hexadecimal

f : floating point

e : floating point (exponent)

c : single character

s : string

% : % character

p : pointer

Ouput Formatting

Output also has formatted specification:

%[flags][width][.precision] type

For long data type, add l at the front of data type:

- long double : (" %lf ")
- unsigned long int : (" %lu ")
- long int. : (" %ld ")

Output Example

- Example 1:

  printf(“%6d”, 34); 34

  printf(”%-6d”, 34); 34....

- Example 2

  printf(“%10s”, "BINUS"); .....BINUS

  printf(“%-10s”, "BINUS"); BINUS.....

  printf(“%8.2f”, 3.14159); 3.14

  printf(“%-8.3f”, 3.14159); 3.141...

Output Operations: putchar() function

Displaying character on the monitor at cursor position. After display, cursor will move to the next position

- Syntax:

  int putchar(int c)

- Header file : stdio.h
- Example :

```
char ch='A';
putchar(ch);
```

Output Operation: putch() function

Display ASCII character to the monitor without moving cursor to its next position

- Syntax:

  int putch(int ch)

- Header file : stdio.h
- Example :

```
char ch='b';
putch(ch);
```

Output Operation: puts() function

Display string to the monitor and move the cursor to new line

- Syntax:

  int puts(const char *str);

- Header file : stdio.h
- Example :

```
puts(”Welcome”);
puts(”to Binus”);
```

Input operation

Standard library function that is related to input operations are:

- scanf();
- getchar();
- gets
- getch();
- getche();

Input operation: function/operation of getting the data into the memory using standard I/O devices (keyboard, disk, etc.)

Input Operation: scanf() function

Display string to the monitor and move the cursor to new line

- Syntax:

  int scanf(constchar *format [,argument]...);

- Header file : stdio.h
- All the argument type are pointers (address of a variable)
- To get the address of a variable use “&”sign

Input Operation: scanf() function

Display string to the monitor and move the cursor to new line

- Example :

```
int aValue;
scanf(”%d”,&aValue);
```

- Input format: ”%type”

BINUS UNIVERSITY

Input Operation: scanf() function

Where type can be substituted with one of the following list:

Type	Used to scan
d **u** **x** **e, f, g** **c**	integer unsigned integer hexadecimal floating point single character
s **o** **[...]** **[^..]**	string ended with white space data unsigned octal string ended with non of the value inside [...] string ended with the value inside [...]

Input Operation: scanf() function

```
/* Program Calculating rectangle area v1*/
#include <stdio.h>
int main(){
    int width, height, area;
    scanf("%d",&width);
    scanf("%d",&height);
    area = width * height;
    return(0);
}
```

```
/* Program Calculating rectangle area v2*/
#include <stdio.h>
int main(){
    int width, height, area;
    scanf(“%d %d”,&width, &height);
    area = width * height;
    return(0);
}
```

Input Operations: getchar() function

Return the next ASCII character from keyboard buffer
Shown on the monitor screen

- Syntax:

```
int getchar(void);
```

- Header file : stdio.h
- Example :

```
char ch;
ch = getchar();
```

Input Operation: getch() function

Return the next ASCII character from keyboard buffer
Does not show on the monitor screen

- Syntax:

```
int getch(void);
```

- Header file : stdio.h
- Example :

```
char ch;
ch = getch(ch);
```

Input Operation: gets() function

read a string from keyboard till find new-line and save in buffer
new-line will later on replaced with null character

- Syntax:

```
char *gets(char *buffer)
```

- Header file : stdio.h
- Example :

```
char buffer[40];
char *ptr;
ptr = gets(buffer);
```

Thank You