



بِنامِ خدا

زبان برنامه نویسی C (21814)

Lecture 2

Chapter 3

مدرس: ناصر سلماسی

زبان برنامه نویسی C (21814)

فصل سوم: مفاهیم اساسی

- زبان C یک زبان برنامه نویسی بسیار قوی
- مورد استفاده برای حل بسیاری از مسایل پیچیده جهان
- امکان برنامه نویسی با یادگیری تعدادی از مفاهیم اساسی

زبان برنامه نویسی C (21814) - فصل سوم

قسمتهای اصلی یک برنامه:

یک برنامه به زبان C شامل

- دستورات اجرایی
- یادداشت های برنامه نویس (comments)
- تعاریف اجزای برنامه
- تعاریف توابع به کار رفته در برنامه

زبان برنامه نویسی C (21814) - فصل سوم

قسمتهای اصلی یک برنامه:

- **Comments :**
 - `/* */`
 - `//`
- **Preprocessor commands**
- **Words**
- **Declaration**
- **Statements**

زبان برنامه نویسی C (21814) - فصل سوم

Preprocessor commands

- دستوراتی که در ابتدای برنامه معرفی و اطلاعات مورد نیاز کامپایلر و مکان آنها را برای پردازش قسمت های مختلف برنامه فراهم می کنند.
- آغاز تمامی **preprocessor command** ها با علامت **#**

زبان برنامه نویسی C (21814) - فصل سوم

Words

- تعریف بسیاری از کلماتی که برای نوشتن یک برنامه به کار می روند، توسط زبان استاندارد برنامه نویسی
- این کلمات به عنوان کلمات کلیدی زبان برنامه نویسی C
- فهرست این کلمه ها در Appendix C کتاب (در وب سایت)
- تعریف سایر کلمات و دستورات مورد نیاز برنامه نویسی توسط کاربر

زبان برنامه نویسی C (21814) - فصل سوم

Declaration

- وظیفه **declaration** ها تعریف یک لغت جدید برای مجموعه کلمه های قابل درک توسط برنامه است.
- هر کلمه ای که در برنامه به کار می رود و متعلق به کلمه های قابل فهم زبان C نباشد، باید توسط کاربر قبل از استفاده در برنامه تعریف شود.

زبان برنامه نویسی C (21814) - فصل سوم

Statements

- یک **statement** همانند یک جمله است و عملی را انجام می دهد.

زبان برنامه نویسی C (21814) - فصل سوم

یک برنامه بسیار ساده:

```
/* This is a very simple Program*/ // Comment
#include <stdio.h> // Preprocessor Commands
// input/output Standard
void main (void) // header line (Function)
{
    puts(" Salam"); // block of codes
    printf(" Hello\n");
}
```

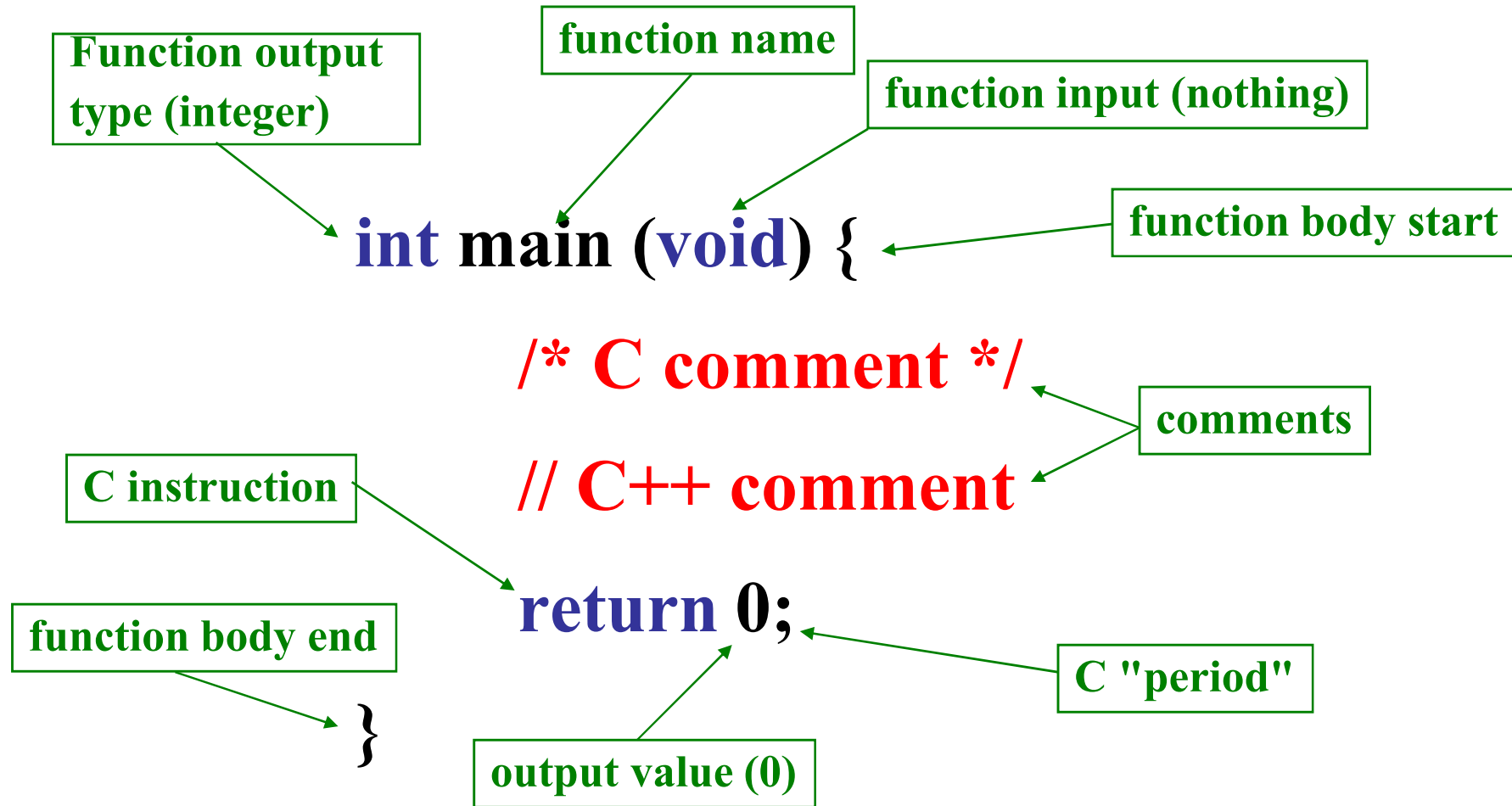
زبان برنامه نویسی C (21814) - فصل سوم

Function ها:

- یک برنامه به زبان C شامل تعدادی **Function** است. عملاً **Function** ها نقطه آغاز اجرای برنامه هستند.
- هر برنامه به زبان C با یک **Function** به نام **main** آغاز می شود.
- **Function** ها داده های ورودی را گرفته، پس از پردازش خروجی ایجاد می کنند.

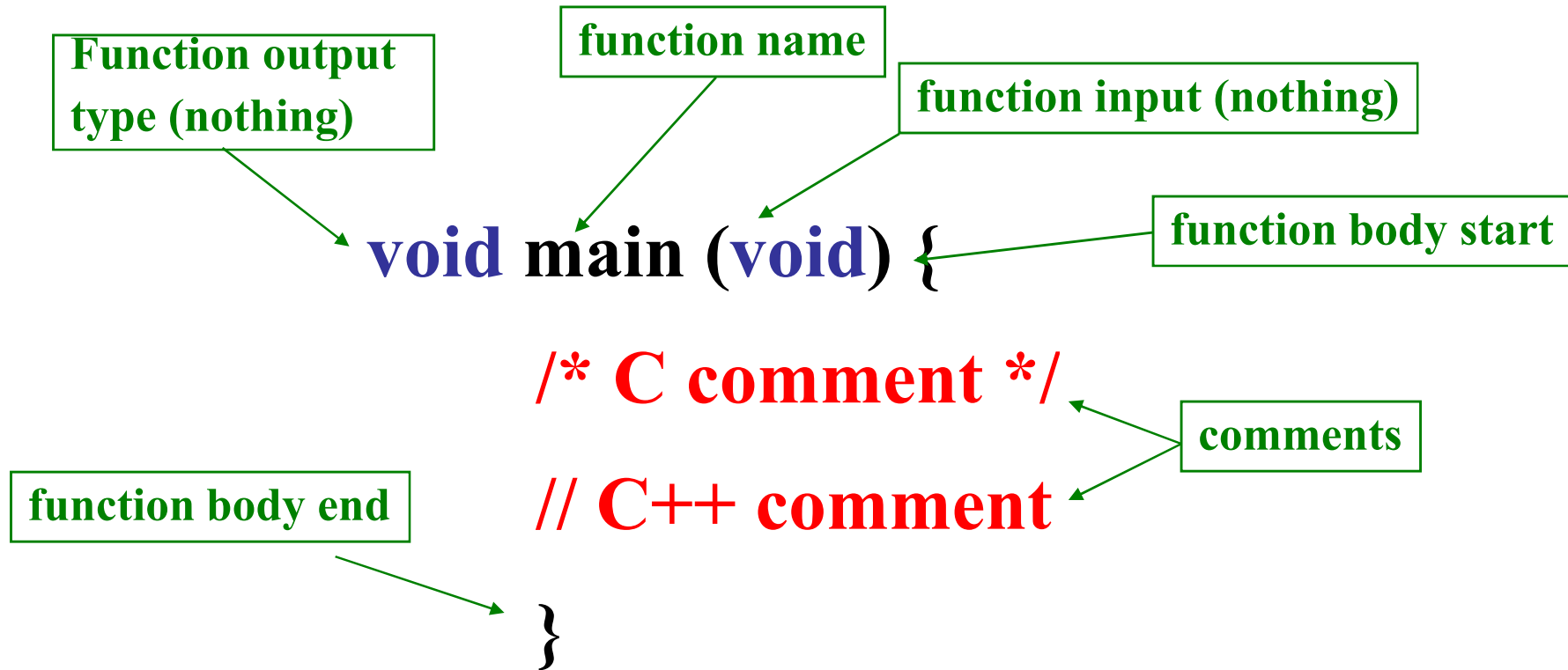
زبان برنامه نویسی C (21814) - فصل سوم

Function Structure



زبان برنامه نویسی C (21814) - فصل سوم

Function Structure



زبان برنامه نویسی C (21814) - فصل سوم

Function ها:

- زبان C شامل تعداد زیادی **Function** آماده است.
- در اغلب برنامه ها کاربر ملزم به نوشتن **Function** هایی برای انجام کارهای مورد نظر خود است.

زبان برنامه نویسی C (21814) - فصل سوم

```
#include <stdio.h>
```

- **Standard I/O library**

زبان برنامه نویسی C (21814) - فصل سوم

void main (void)

- **Standard I/O library**

- گاهی توابع هیچ مقداری را در جایی که صدا زده می شوند بر نمی گردانند. معمولاً هدف از بکار بردن چنین توابعی ایجاد تاثیرات جانبی مانند تغییر دادن مقدار یک متغیر می باشد. به چنین توابعی توابع **void** گفته می شود زیرا در ابتدای آنها از کلمه **void** استفاده می شود.

زبان برنامه نویسی C (21814) - فصل سوم

- **Strings** کلماتی هستند که میان " " ها قرار می گیرند.

زبان برنامه نویسی C (21814) - فصل سوم

متغیر ها، ورودی ها، خروجی ها و ترتیب آنها:

- همانطور که بیان شد، هدف از اجرای یک برنامه، انجام پردازش هایی به روی داده ها است.
- پیش از آغاز نوشتن یک برنامه بایستی بدانیم که اطلاعات مورد نیاز را از کجا و چگونه بدست آوریم، چگونه آنها را ذخیره و استفاده کنیم و چگونه پاسخ ها را گزارش دهیم.

زبان برنامه نویسی C (21814) - فصل سوم

متغیرها (Variables):

- یک متغیر قسمتی از حافظه کامپیوتر است که توسط برنامه به آن نامی داده شده و برای ذخیره سازی اطلاعات و بازیابی آن بکار می رود.
- متغیر می تواند هر نامی داشته باشد به جز نام دستورات قابل درک زبان C.
- یکسری دستورالعمل ها برای انتخاب نام متغیر وجود دارد.
- برنامه نویس متبحر سعی در انتخاب نام مناسب و کوتاه دارد.

زبان برنامه نویسی C (21814) - فصل سوم

متغیر ها (ادامه):

- هر متغیر دارای نوع (Type) می باشد (عددی یا حرفی). میزان فضای اختصاص یافته برای هر متغیر بستگی به نوع آن دارد.
- هر متغیر با **Declaration** تعریف میشود. متغیر ها معمولا در ابتدای برنامه تعریف می شوند. بهتر است پس از **Preprocessor Commands** ها تعریف شوند.

```
int Minute;  
double Second;
```

زبان برنامه نویسی C (21814) - فصل سوم

متغیر ها (ادامه):

- برای تعریف هر متغیری کامپایلر فضایی را برای متغیر تخصیص می دهد. این فضای اختصاص یافته به نام متغیر مرتبط می شود.
- گاهی اوقات در برنامه نویسی مجبور به رجوع به آدرس متغیر می شویم که در چنین حالتی از [ampersand](#) در جلوی نام متغیر استفاده می کنیم. **&Minute**

زبان برنامه نویسی C (21814) - فصل سوم

نحوه اختصاص مقادیر به متغیر ها:

```
int age;      //declaration of integer variable age  
age = 20;     //assignment of value 20
```

or

```
int age = 20; //declaration & assignment
```

زبان برنامه نویسی C (21814) - فصل سوم

ورودی ها و خروجی ها:

فرمت های ورودی و خروجی جزو پیچیده ترین قسمت های برنامه نویسی C می باشند. ساده ترین آنها عبارتند از:

نام دستور	مفهوم	کارکرد
<code>puts()</code>	نمایش <code>string</code>	نوشتن پیام به روی خروجی
<code>printf()</code>	چاپ خروجی با بکار گیری فرمت	نوشتن پیام و اطلاعات به روی خروجی
<code>scanf()</code>	دریافت ورودی با بکار گیری فرمت	دریافت اطلاعات از صفحه کلید

زبان برنامه نویسی C (21814) - فصل سوم

- این توابع همگی در **stdio library** تعریف شده اند که با نوشتن آن در ابتدای برنامه به برنامه اضافه شده اند.

- تفاوت میان **puts** و **printf**

- بکارگیری فرمت های ورودی:

```
scanf ("%i", &minutes);
```

```
scanf ("%i %i", &age, &weight);
```

زبان برنامه نویسی C (21814) - فصل سوم

بکارگیری فرمت های خروجی:

- معمولاً فرمت های خروجی پیچیده تر هستند.

```
Age = 21;
```

```
Weight = 65;
```

```
printf("Age: %i Weight: %i \n", Age, weight);
```

```
Age: 21 Weight: 65
```


زبان برنامه نویسی C (21814) - فصل سوم

بکارگیری فرمت های خروجی:

\t •

\n •

\n\n\n •

" " •

زبان برنامه نویسی C (21814) - فصل سوم

مثال:

```
printf("one\ntwo"); prints
```

one

two

```
printf("one\ttwo"); prints
```

one two

```
printf("one two three"); prints
```

one two three

زبان برنامه نویسی C (21814) - فصل سوم

مثال:

```
/* This Program Reads and echoes a single number .*/  
#include <stdio.h>  
void main (void){  
    int number; //to store the Number  
  
    puts( "Please type a single number and hit enter");  
    scanf( "%i" ,&number);  
    printf("You typed %i.\n",number);  
}
```

زبان برنامه نویسی C (21814) - فصل سوم

Print Specifiers and Parameters

- با استفاده از دستور `printf()` می توان در یک زمان چندین خروجی داشت:

```
printf(" %s\n%s", "one", "two");
```

Diagram illustrating the components of the `printf()` function call:

- The string `" %s\n%s"` is labeled as **print specifiers**.
- The string `"one"` is labeled as **parameter one**.
- The string `"two"` is labeled as **parameter two**.
- The string `"two"` is also labeled as **parameter three**.

which prints:

one

two

زبان برنامه نویسی C (21814) - فصل سوم

تمرین:

برنامه ای بنویسید که شماره دانشجویی شما را گرفته و سپس به روی خروجی چاپ کند.

زبان برنامه نویسی C (21814) - فصل سوم

The Big Picture

- **The source of a C-Program is a text file – called *source code*.**
- **The compiler reads the source and reports errors.**
- **Execution starts at beginning of *main()* function.**
- **The program executes line by line.**

زبان برنامه نویسی C (21814) - فصل سوم

Some Details

- **C is case sensitive (printf not same as Printf).**
- ***White Space* in the source code is ignored by the compiler (program).**
 - Spaces, tabs, newline (Enter key).
- **Things have names.**
 - include, main, printf, return.
- **Punctuation, other symbols have specific meaning.**
 - # ; , <> { } %

زبان برنامه نویسی C (21814) - فصل سوم

تمرین شماره یک

- نحوه نام گذاری فایل
- **Comment** نویسی و **Indentation** فراموش نشود.

زبان برنامه نویسی C (21814) - فصل سوم

محاسبات ساده

- پس از آموزش نحوه ورود و خروج داده ها از حافظه کامپیوتر، نحوه انجام محاسبات ساده بیان می شود.
- اعداد: اعداد را می توان به چندین صورت در حافظه کامپیوتر ذخیره نمود.
- اعداد صحیح (integer Type) که برای ذخیره اعداد صحیح بکار می رود. (این اعداد می تواند مثبت یا منفی باشند).

int Numbers;

- اعداد اعشاری (double Type) که برای ذخیره اعداد اعشاری بکار می رود.

double Ratio;

زبان برنامه نویسی C (21814) - فصل سوم

مثال:

```
double Ratio;  
printf ("Please enter the Ratio:");  
scanf( "%lg", &Ratio);  
printf( "You entered %g \n", Ratio);
```

زبان برنامه نویسی C (21814) - فصل سوم

روشهای تخصیص یک مقدار به یک متغیر:

- دریافت مقدار از طریق User با دستور `scanf`
- اختصاص مقدار در داخل برنامه

Ratio = 5;

زبان برنامه نویسی C (21814) - فصل سوم

مقادیر ثابت (Constants):

- **Constant** ها در داخل برنامه مقادیری هستند که یکبار در ابتدای برنامه تعریف شده و تغییر نمی کنند. روش تعریف آنها بشرح زیر است:

```
#define PI 3.1416
```

- در هنگام استفاده از دستور **define** توجه به نکات زیر ضروری است:

- No semicolon
- No = sign

زبان برنامه نویسی C (21814) - فصل سوم

فرمول های محاسباتی:

- فرمول های محاسباتی، که **expressions** نامیده می شوند، به راحتی نوشته می شوند:

$$\text{Area} = \text{PI} * r * r ;$$

زبان برنامه نویسی C (21814) - فصل سوم

مثال:

محاسبه مجموع و میانگین سه عدد:

```
/* This is a code to calculate the summation as well as the average of three
arbitrary numbers */
#include <stdio.h>
#define Multiplier 10

void main (void){
    double A,B,C;           // Define three variables to get the values
    double Sum, Average;

    printf(" Please enter three numbers\n");
    scanf(" %lg%lg%lg", &A,&B,&C);
    Sum= A+B+C;
    Average=Sum/3;

    printf("\n The Summation of the numbers is %g\n",Sum);
    printf("\n The Average of the numbers is %g\n",Average);
}
```

زبان برنامه نویسی C (21814) - فصل سوم

مثال:

محاسبه مساحت یک دایره



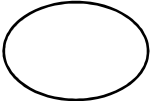
زبان برنامه نویسی C (21814) - فصل سوم

تمرین:

برنامه ای بنویسید که مربع و مکعب اعداد 1،2،3،4 را محاسبه و نمایش دهد.

زبان برنامه نویسی C (21814) - فصل سوم

نمودار جریان (Flow Chart)

	عملیات
	وجود خروجی های متفاوت در اثر مقایسه
	آغاز و خاتمه برنامه

زبان برنامه نویسی C (21814) - فصل سوم

حالت های شرطی:

- یکی از مزایای اصلی استفاده از کامپیوتر قابلیت آن در انجام واکنش های مختلف در حالت های متفاوت است.
- یکی از این روشها استفاده از دستورات `if` و `else if` است.

زبان برنامه نویسی C (21814) - فصل سوم

:if Statement

- در یک if Statement ساده پس از کلمه شرطی **if**، شرط در داخل پرانتز بیان می شود.
- پس از پرانتز در داخل { } دستوراتی که در صورت صحیح بودن شرط بایستی اجرا شوند، نوشته می شوند.
- در هنگام اجرای برنامه، در صورتیکه شرط برنامه صحیح باشد، دستورات داخل { } اجرا می شوند. در غیر اینصورت نادیده گرفته می شوند.

زبان برنامه نویسی C (21814) - فصل سوم

```
/* This Program Reads and echoes a single number. */
```

```
#include <stdio.h>
```

```
void main (void){
```

```
    int number;    //to store the Number
```

```
    printf( "Please type a single number: ");
```

```
    scanf( "%i" ,&number);
```

```
    if(number >100){
```

```
        printf("\n\n You typed %i.\n",number);
```

```
        printf("\n\n You typed a number greater than 100.\n");
```

```
    }
```

```
    printf("\n Have a good day.\n");
```

```
}
```

زبان برنامه نویسی C (21814) - فصل سوم

:if ... else Statement

- اگر مایل باشیم در صورت درست نبودن شرط، دستورات دیگری اجرا شوند، از عبارت فوق استفاده می کنیم.

زبان برنامه نویسی C (21814) - فصل سوم

/* This Program Reads and echoes a single number. */

#include <stdio.h>

void main (void){

int number; //to store the Number

printf("Please type a single number: ");

scanf("%i" ,&number);

if(number >100){

printf("\n\n You typed %i.\n",number);

printf("\n\n You typed a number greater than 100.\n");

}

else{

printf("\n\n You typed %i.\n",number);

printf("\n\n You typed a number less than 100.\n");

}

printf("\n Have a good day.\n");

}

زبان برنامه نویسی C (21814) - فصل سوم

if ... else Statement (ادامه):

- عبارت if ... else می تواند بصورت مکرر در صورت وجود شرط های مختلف به کار رود.
- اگر فقط بخواهیم یک خط را در صورت صحیح بودن گزاره if اجرا کنیم، می توانیم براکت ها { } را حذف کنیم.

زبان برنامه نویسی C (21814) - فصل سوم

```
#include <stdio.h>
void main (void){
    int number;        //to store the Number

    printf( "Please type a single number: ");
    scanf( "%i" ,&number);

    if(number >100){
        printf("\n\n You typed %i.\n",number);
        printf("\n\n You typed a number greater than 100.\n");
    }
    else if (number<100){
        printf("\n\n You typed %i.\n",number);
        printf("\n\n You typed a number less than 100.\n");
    }
    else{
        printf("\n\n You typed %i.\n",number);
        printf("\n\n You typed a number equal to 100.\n");
    }
    printf("\n Have a good day.\n");
}
```


زبان برنامه نویسی C (21814) - فصل سوم

مثال:

برنامه ای بنویسید که درجه حرارت هوا را بصورت فارنهایت از کاربر گرفته و پس از محاسبات لازم اعلام نماید که درجه حرارت به مقیاس سانتی گراد بزرگتر، مساوی، و یا کوچکتر از صفر می باشد.

زبان برنامه نویسی C (21814) - فصل سوم

حلقه ها و تکرار ها:

- یکی از مزایای برنامه نویسی استفاده از لوپ (حلقه های تکرار) می باشد که اجازه می دهد که گروهی از دستورات را چندین مرتبه تکرار کنیم. این حلقه ها برنامه نویسی را ساده می کنند.

- زبان برنامه نویسی C شامل سه نوع حلقه می باشد:

– while

– do

– for

زبان برنامه نویسی C (21814) - فصل سوم

/* This Program fills a screen with number lines. */

#include <stdio.h>

void main (void){

int number; //to store the Number

printf("Please type a single number: ");

scanf("%i" ,&number);

while(number>0){

printf(" %i \t",number);

number=number-1;

}

printf("\n The code is terminated\n");

}

زبان برنامه نویسی C (21814) - فصل سوم

دستور **do ... while**

فرم عمومی این دستور به شرح زیر است:

```
do{  
    statements  
}while (expression)
```

- حلقه **do** اجرای **statement** ها را تا هنگامی که شرط **expression** صحیح باشد تکرار می کند.

- مثال: طراحی یک تست چهار جوابی

زبان برنامه نویسی C (21814) - فصل سوم

```
/* This Program asks a question */
#include <stdio.h>
#include <conio.h>
void main (void){
    char Answer ;
    printf ( "Please Answer this question:\n\n ");
    printf ( "What is the capital of Iran?\n\n ");
    printf ( " A: Isfahan\n B: Tabriz\n C: Mashhad\n D: Tehran\n\n");

    do{
        Answer = getche();
    }while( Answer!='A' && Answer!='B' && Answer!='C' && Answer!='D' );

    printf("\n You have answered the question.\n\n");

    if(Answer=='D'){
        printf(" **** Your answer is correct. Good job. ****\n\n");
    }
    else{
        printf(" **** Your answer is wrong!!! Study more ****\n\n");
    }
}
```

زبان برنامه نویسی C (21814) - فصل سوم

حلقه for:

حلقه for تقریبا همان وظایف حلقه های قبلی را بصورت متفاوتی انجام می دهد.

مثال: محاسبه مربع و مکعب اعداد 1 الی 4

زبان برنامه نویسی C (21814) - فصل سوم

```
#include <stdio.h>
void main (void){

    int number, Square, Cube;    //to store the Number
    int i;
    printf( "Please type a single number: \n\n");
    scanf( "%i" ,&number);

    for (i = 1; i <= number; i++){
        Square = i*i;
        Cube = Square *i;
        printf( " The Square and Cube of %i\t are %i\t and %i,
respectively.\n" ,i, Square, Cube);

    }
}
```

زبان برنامه نویسی C (21814) - فصل سوم

تمرین:

برنامه ای بنویسید که در آن اعداد زوج در فاصله 1 الی 1000 در خروجی کامپیوتر چاپ شود.

زبان برنامه نویسی C (21814) - فصل سوم

تمرین:

برنامه ای بنویسید که در آن از میان یکسری اعداد داده شده توسط کاربر (تعداد آنها توسط کاربر تعیین می شود)، مقادیر حداکثر و حداقل آنها را محاسبه و چاپ نماید.

زبان برنامه نویسی C (21814) - فصل سوم

تمرین:

برنامه ای بنویسید که دترمینان یک ماتریس مربع 3×3 را محاسبه نماید.