دانشگاه بین المللی امام خمینی قزوین

دانشکدهٔ فنی و مهندسی

مبانی کامپیوتر و برنامه سازی

(دومین ترم کرونا!)

فصل چهارم: مقدمات برنامه سازی به زبان C

نستوه طاهری جوان

nastooh@aut.ac.ir

نرم افزار

یک تعریف برای نرم افزار:

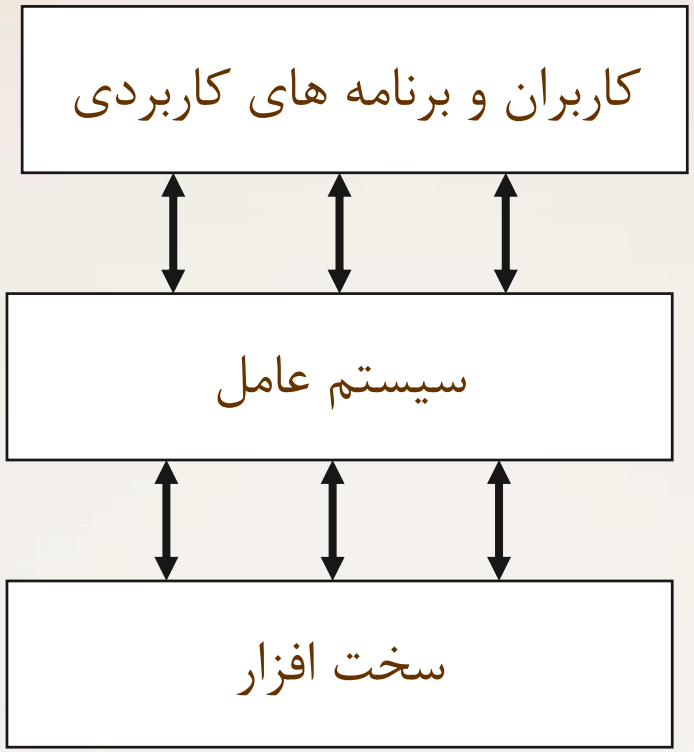
- برنامه: بیان الگوریتم ها به کمک یک زبان
- نرم افزار: مجموع یک یا چند برنامه با هدف خاص

- انواع نرم افزار:
  - نرم افزارهای کاربردی: برای رفع نیازهای معمول، مانند دیکشنری، بازی، حسابداری، انبارداری، دانشگاهی و و و
  - نرم افزارهای سیستمی: بهره برداری از سخت افزار یا سایر نرم افزارها، مانند سیستم عامل و کامپایلر...

nastooh@aut.ac.ir

نرم افزار

- ✓ جایگاه سیستم عامل
  - سیستم عامل: مهم ترین نرم افزار سیستمی و واسط بین سخت افزار و انسان



nastooh@aut.ac.ir

نرم افزار

- ✓ زبان ماشین
  - زبانی که ماشین آن را درک میکند، فقط صفر و یک است.
  - برنامه های اولیه تماماً به زبان ماشین نوشته می شدند.
  - این زبان برای انسان طاقت فرسا و مشکل است.

- ✓ زبان برنامه سازی
  - جهت راحتی کار برنامه نویسان، زبانهایی به غیر از زبان ماشین ابداع شدند.

مترجم ها وظیفهٔ ترجمهٔ برنامه های نوشته شده به سایر زبانها به زبان ماشین را برعهده دارند.

nastooh@aut.ac.ir

زبان های برنامه سازی

✓ انواع زبان های برنامه سازی

- سطح پایین
  - نزدیک به زبان ماشین
  - نیاز به آشنایی عمیق با معماری کامپیوتر
  - انعطاف و قدرت بالاتر
  - مانند اسمبلی
- سطح میانی
  - ترکیب سادگی زبانهای سطح بالا و انعطاف و قدرت سطح پایین
  - مانند C/C++
- سطح بالا
  - نزدیک به زبان محاوره انسان
  - ترجمهٔ یک جملهٔ آن به چندین جملهٔ زبان ماشین
  - فراگیری راحت تر، عدم نیاز به آشنایی عمیق با معماری کامپیوتر
  - مانند پاسکال، بیسیک

nastooh@aut.ac.ir

نرم افزار

- ✓ مترجم
  - وظیفه: تبدیل برنامه های نوشته شده به زبان برنامه سیازی به زبان ماشین
  - انواع:
    - مفسر
      - تبدیل خط به خط به زبان ماشین
      - مانند مفسر زبان بیسیک
    - کامپایلر
      - تبدیل یکجا به زبان ماشین
      - مانند کامپایلر زبان C

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ تاریخچهٔ زبان C

رده: زبانهای برنامه نویسی ساخت-یافته

- ۱۹۷۲: ابداع زبان C توسط دنیس ریچیِ فقید با الهام از زبان B در آزمایشگاه بل
- ۱۹۷۰: ابداع زبان B توسط کن تامپسون با الهام از زبان BCPL
- ۱۹۶۷: ابتداع زبان BCPL توسط مارتین ریچاردز

- هدف اولیه: طراحی سیستم عامل یونیکس

- استقبال بی سابقه از این زبان در دهه های ۸۰ و ۹۰ میلادی و تداوم آن تا اکنون!

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ چند نکته اولیه

- زبان C اصطلاحاً حساس به متن (Case Sensitive) است.
  - حروف کوچک و بزرگ، متفاوتند!
- استفاده از توابع کتابخانه ای در C اکیداً توصیه می شود.
  - قطعه برنامه هایی که سازندگان C برای استفادهٔ عموم نوشته اند.
- C یک زبان همه منظوره (General Purpose) است.
  - تقریباً همه نوع برنامه ای می توان با C نوشت. مانند بازی، برنامه های مهندسی و و و
- استفاد از یک IDE مناسب، کمک شایانی به راحتی برنامه نویسی می کند.
  - مثال: Microsift Visual Studio، یا Code::Blocks، یا Eclipse و و و

nastooh@aut.ac.ir

برنامه نویسی به زبان C

- ساختار برنامه های C
  - یک برنامه نمونه:

```
#include<نام فایل های سرآیند>
void main( )
{

    دستورات اجرایی برنامه

}
```


nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ ساختار برنامه های C

○ چند نکته:

- بدنهٔ اصلی برنامه، تابعی به نام main است.
- توابعی از قبل نوشته شده اند و جهت راحتی کار برنامه نویس همراه کامپایلر سی عرضه می شوند. (موسوم به توابع کتابخانه ای زبان C)
  - ➢ می توان تعبیر کرد، استفاده از این توابع برای جلوگیری از اختراع دوباره چرخ است!!
  - ➢ اطلاعات این توابعِ از پیش نوشته شده، در تعدادی فایل با نام فایل سرآیند قرار دارند.
  - ➢ پسوند فایلهای سرآیند معمولا h. هستند.
  - ➢ قبل از استفاده از این توابع باید دقیقا بدانیم هر تابع در کدام فایل سرآیند است.
  - ➢ برای استفاده از یک تابع، باید فایل سرآیند آن را با دستور #include به ابتدای برنامه اضافه کنیم.
- بعد از هر دستور در زبان سی، باید سمی کالن (;) گذاشته شود.

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ انواع داده در زبان C

1. char: یک بایت است. محدوده: ۱۲۸- تا ۱۲۷
2. int: دو بایت است. محدوده: ۳۲۷۶۸- تا ۳۲۷۶۷
3. float: ۴ بایت. برای اعداد اعشار تا حدود ۷ رقم دقت.
4. double: ۸ بایت. برای اعداد اعشار، تا حدود ۱۶ رقم دقت.
5. void: در جای خود بحث خواهد شد.

- **نکته مهم**: طول داده ها ممکن است بر اساس نوع کامپایلر و نوع سیستم عامل تغییر کند. به عنوان مثال در بسیاری از پیاده سازی های امروزه، نوع int به صورت ۴ بایتی است. یعنی بازهٔ 2,147,483,648- تا 2,147,483,647

nastooh@aut.ac.ir

برنامه نویسی به زبان C

- ✓ انواع داده در زبان C
  - پیشوندهای تغییر دهندهٔ نوع

1. signed: نوع را علامتدار می کند. (پیش فرض همه علامتدار هستند!!!)
2. unsigned: نوع را بدون علامت می کند.
3. short: طول دادهٔ نوع را نصف می کند.
4. long: طول دادهٔ نوع را دو برابر می کند.

- جلوی int و char هر چهار پیشوند استفاده می شوند.
- جلوی float و double فقط short و long استفاده می شوند.

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ انواع داده در زبان C

○ چند نکته:

- برخی از زبان ها برای داده های منطقی (Boolean) نوع دادهٔ خاص دارند. در C اینگونه نیست.
- برخی از زبان ها برای داده های رشته ای (String) نوع دادهٔ خاص دارند. در C اینگونه نیست.
- اعداد **صحیح** را در زبان C می توان هم در مبنای ۱۰ و هم مبنای ۸ و هم مبنای ۱۶ نمایش داد. برای نمایش مبنای ۸، یک 0 و برای نمایش مبنای ۱۶ یک 0X قبل از عدد نمایش داده می شود.

  23 027 0x17

- مقادیر کاراکتری در زبان C مابین تک گیومه نمایش داده می شوند

  'A'

- مقادیر رشته ای در زبان C مابین گیومهٔ دوتایی نمایش داده می شوند.

  "Ali"

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ متغیر در زبان C

- تعریف متغیر:
  - در واقع نامی برای کلمات حافظه که می تواند در طول برنامه تغییر کند.
- نامگذاری متغیر:
  - ترکیبی از حروف a تا z، A تا Z، ارقام و _ (کاراکتر خط زیر)
  - اولین حرف نام متغیر **نمی تواند** رقم باشد.
  - نام متغیر نمی تواند از کلمات کلیدی زبان C باشد. (حدود ۲۰ کلمه کلیدی داریم)
- تعریف متغیر:
  - قبل از استفاده باید متغیر را تعریف کرد. نحوه انجام کار:

```
نوع داده   نام متغیر ;

char  c;
long  int  i;
```


nastooh@aut.ac.ir

برنامه نویسی به زبان C

- ✓ متغیر در زبان C
  - مقدار دهی به متغیر:
    - هنگام تعریف (به آن initialize کردن گویند)
    - پس از تعریف و با دستور انتساب
    - با دستورات ورودی

```
int  i = 61,  j = 20;
char  c = 'q';
long int  ave,  count_3 = 1,  _sum = 0;
….
count = 54;
```


nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ ثابت ها در زبان C

○ با استفاده از const

- نوع نیاز دارد.
- سمی کالن نیاز دارد.
- حافظه اشغال می کند.

const نوع ثابت نام ثابت = مقدار ثابت ;

مثال:

```
const int I = 100;
```

○ با استفاده از #define

- نوع نیاز ندارد.
- حافظه اشغال نمی کند.
- سمی کالن نمی خواهد.

#define نام ثابت مقدار ثابت

مثال:

```
#define PI 3.14
```


nastooh@aut.ac.ir

برنامه نویسی به زبان C

- توصیه برنامه نویسی

- برای متغییرها و ثوابت بهتر است از نام های معنی دار استفاده کنیم.
  - این کار به خوانایی برنامه برای دیگران (و حتی خودمان) کمک می کند.

- معمولا نام متغییرها را با حروف کوچک مشخص می کنیم.

- برای جدا کردن قسمت های مختلف یک نام، از خط زیر (_) استفاده می کنیم.
  my_salary discount_value

- معمولاً نام **ثابت** را با حروف بزرگ تایپ می کنیم.
  - این کار اجباری نیست و فقط به خوانا تر شد کد کمک می کند.

- با استفاده از const

nastooh@aut.ac.ir

برنامه نویسی به زبان C

- ✓ عملگرها در زبان C
  - جهت انجام عملیات بر روی مقادیر و متغیرها
  - انواع عملگرها (Operators)
    1. انتساب
    2. محاسباتی
    3. رابطه ای
    4. منطقی
    5. بیتی
    6. انتسابی ترکیبی
    7. یکانی
    8. سایر عملگرها

برنامه نویسی به زبان C

- ✓ عملگرها در زبان C
  - ○ عملگر انتساب: علامت =

- مقدار سمت راست را در متغیر سمت چپ کپی می کند.

```
i = 54;
```


nastooh@aut.ac.ir

برنامه نویسی به زبان C

- ✓ عملگرها در زبان C
  - عملگرهای محاسباتی
    - جمع: +
    - تفریق: -
    - ضرب: *
    - تقسیم: /
    - باقیمانده: %

مثال:

```
int  i,  j = 5,  k;
k = 4 + i * 5;
…
k = j / 2;
```

مقدار k در نهایت چند است؟ 2 یا 2.5؟

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای محاسباتی

- چند نکته:

➢ عملگر **تقسیم** هم اعشاری و هم صحیح عمل می کند. اگر هر دو طرف صحیح باشد، حاصل صحیح است. اما اگر یکی از طرفین اعشاری باشد، حاصل اعشاری است.

« مثال: مقدار نهایی متغیر k در دو حالت زیر چند است؟ دقت کنید در هر دو حالت، متغیر k از نوع float، یعنی اعشاری است.

```
int  i = 2,  j = 5;
float  k;
k = j / i;
```

```
int  j = 5;
float  k,  i = 2;
k = j / i;
```

➢ **تقدم عملگرهای محاسباتی:**

اولویت * و / و % همواره بالاتر از + و – است و جهت بکار بستن آنها در یک عبارت از چپ به راست است.

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ عملگرها در زبان C

- توصیه برنامه نویسی:
  - بهتر است قبل و بعد از عملگرها، یک فاصله (فقط جهت افزایش خوانایی) تایپ کنید.
    - این کار الزامی نیست، اما به خواناتر شد کد شما کمک می کند.

```
c=a+b*c/d;

c = a + b * c / d;
```

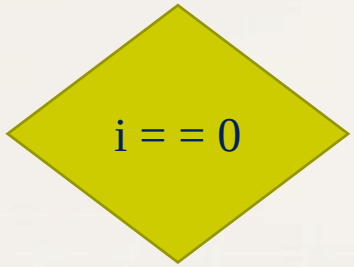

nastooh@aut.ac.ir

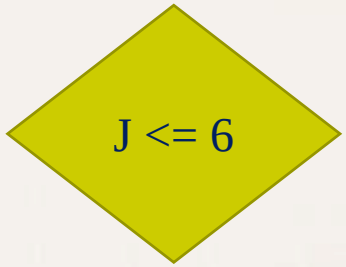
برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای رابطه ای (مقایسه ای)

- برای مقایسه دو مقدار بکار می روند. در واقع ارتباط دو عملوند خود را مشخص می کنند.
- شامل: < و =< و > و => و = = و =!
- اگر حاصل یک عبارت منطقی یا برابر true است یا برابر false.
- در عمل اگر حاصل برابر "درست" بود، مقدار آن یک و اگر حاصل آن برابر "نادرست" بود، مقدار آن صفر است.
- مثال؛ استفاده در فلوچارت:





nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای رابطه ای (مقایسه ای)

- اولویت عملگرهای رابطه ای:

➢ اولویت عملگرهای < و =< و > و => همواره از دو عملگر = = و =! بالاتر است.

➢ جهت بکار بستن همگی، از چپ به راست در یک عبارت است.

اولویت	عملگر	جهت بکار بستن
1	=> و > و =< و <	⟶
2	= = و =!	⟶

- هنگام برنامه نویسی، مراقب شباهت ظاهری عملگرهای = و = = باشید!!!

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای منطقی

- بر روی عبارت منطقی و رابطه ای عمل می کنند.
- برای ساختن عبارات پیچیده تر
- شامل:

1. یا (or) منطقی: ||
2. وَ (and) منطقی: &&
3. نقیض (Not) منطقی: !

(a= =b) || (a= =0)

(8 > 3) && (4 < 7)

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای منطقی

- حاصل عملگرهای منطقی:

A	B	!A	!B	A || B	A && B
T	T	F	F	T	T
T	F	F	T	T	F
F	T	T	F	T	F
F	F	T	T	F	F

- اولویت عملگرهای منطقی: اولویت ! بالاتر از && است و اولویت && بالاتر از || است.
- نحوهٔ به کار بستن && و || از چپ به راست است. ولی نحوهٔ به کار بستن ! **استثناً** از راست به چپ است.

nastooh@aut.ac.ir

برنامه نویسی به زبان C

- ✓ عملگرها در زبان C
  - ○ عملگرهای منطقی
    - اولویت عملگرهای منطقی و رابطه ای در کنار هم

اولویت	عملگر	جهت بکار بستن
1	!	←
2	<= و < و >= و >	→
3	!= و ==	→
4	&&	→
5	||	→

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای بیتی

- برای دستکاری بیتیِ عملوندهای از نوع **صحیح** به کار می روند.
- شامل:
  - and بیتی: &
  - or بیتی: |
  - xor بیتی: ^ که به آن یای انحصاری (Exclusive Or) نیز گویند.
  - not بیتی: ~
  - شیف راست: >>
  - شیفت چپ: <<

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای بیتی

- نحوهٔ عملیات عملگرهای بیتی:

A	B	~A	~B	A | B	A & B	A ^ B
0	0	1	1	0	0	0
0	1	1	0	1	0	1
1	0	0	1	1	0	1
1	1	0	0	1	1	0

- مثال:

```
a = 24;
b = 53;
x = a & b ;
y = a  | b ;
z = a  ^  b ;
w = a << 2;
m = ~b;
```

a :	0	0	0	1	1	0	0	0	24
b :	0	0	1	1	0	1	0	1	53
x:	0	0	0	1	0	0	0	0	16
y :	0	0	1	1	1	1	0	1	61
z:	0	0	1	0	1	1	0	1	45
w:	0	1	1	0	0	0	0	0	96
m:	1	1	0	0	1	0	1	0	-54

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای بیتی

- چند نکته:

➢ به هنگام هر شیفت، به تعداد نیاز صفر از سمت مقابل افزوده می شود.

➢ هر شیف به راست مانند این است که عدد را بر ۲ تقسیم کرده باشیم.

➢ هر شیفت به چپ مانند این است که عدد را در ۲ ضرب کرده باشیم. (تا زمانی که سرریزی نداشته باشیم)

➢ اولویت عملگرهای بیتی:

اولویت	عملگر	جهت بکار بستن
1	~	←
2	<< و >>	→
3	&	→
4	^	→
5	|	→

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای یکانی

- دسته ای از عملگرهای دیگر هستند که همگی یک عملوند دارند.
- اولویت این عملگرها از همه بیشتر است.
- شامل:
  - + (علامت مثبت)
  - - (علامت منفی)
  - ++ (خوانده می شود: پلاس پلاس یا increment)
  - - - (خوانده می شود: ماینس ماینس یا decrement)
  - * (مربوط به اشاره گر ها)
  - & (مربوط به اشاره گر ها)
  - sizeof
  - (تبدیل نوع)

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای یکانی

- نکته مهم در مورد عملگر های ++ و – –:

➢ میتوان گفت: a++; و یا ++a; همانند عبارت روبروست: a = a + 1;

➢ این دو عملگر هم قبل از عملوند خود می آیند و هم بعد از آن.

➢ این دو عملوند هم به تنهایی استفاده می شوند، و هم در یک عبارت.

➢ اگر این عملگرها در عبارتی ظاهر شود (همراه با عملگر های دیگر) فرم پیشوندی و پسوندی آن متفاوت است. در حالت پیشوندی، ابتدا عملگر بر روی عملوند عمل کرده و سپس عملوند را با مقدار جدید به عبارت میفرستد. اما در حالت پسوندی ابتدا عملوند را در عبارت میفرسد، سپس آن را افزایش یا کاهش می دهد!

```
a = 5;
a++;
++a;
```

```
a = 5;
b = a++;
```

```
a = 5;
b = ++a;
```

nastooh@aut.ac.ir

برنامه نویسی به زبان C

- ✓ عملگرها در زبان C
  - ○ عملگرهای یکانی
    - مثال:

```
a = 10;
b = ++a  +  ++a;
```

طبق تعریف زبان C در این حالت متغیر b برابر ۲۴ خواهد بود و در نهایت a برابر با ۱۲.

- مثال:

```
a = 10;
b = a++  +  a++;
```

طبق تعریف زبان C در این حالت متغیر b برابر ۲۰ خواهد بود و در نهایت a برابر با ۱۲.

در برخی کامپایلرها، مانند Code::Blocks، عمگرهای ++ و -- پیشوندی و پسوندی اگر بیش از یکبار در یک عبارت ظاهر شوند، به فرمت دیگری پیاده سازی شده اند!!! از بکار بردن اینگونه عبارات در برنامه های خود پرهیز کنید!

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ دیگر عملگرها

- ?:

➢ نحوه بکار بستن:

متغیر = شرط ? عبارت یک : عبارت دو ;

اگر شرط برقرار بود، عبارت یک و اگر نبود عبارت دو در متغیر قرار می گیرد!

مثال

```
a = (x > y) ? c : b ;
```

- ,

➢ برای بکار بردن چند دستور در یک عبارت.

- ()

➢ پرانتز اولویت عملگرها را تغییر می دهد.

برنامه نویسی به زبان C

✓ عملگرها در زبان C

○ عملگرهای انتسابی-ترکیبی

- ترکیب عملگر انتساب بهمراه عملگرهای دوتایی محاسباتی و بیتی

➢ شامل: =- و =+ و =/ و =% و =* و =& و =| و =^ و =<< و =>>

➢ این عملگرها فقط برای کوتاه نویسی هستند.

➢ مثال:

a += 5;	**b *= c;**	**d <<= 2;**
معادل	**معادل**	**معادل**
a = a + 5;	**b = b * c;**	**d = d << 2;**

➢ اولویت همهٔ عملگرهای انتسابی ترکیبی یکسان است و جهت به کار بستن آنها در یک عبارت از راست به چپ است.

```
a = b = c = d = 9;
```

برنامه نویسی به زبان C

✓ عملگرها در زبان C

1	( ) و [ ] و -> و .	
2	sizeof( ) و & و ~ و ! و (پیش وندی)-- و (پیش وندی)++ و (علامت منفی)–	←
3	* و / و %	
4	+ و -(تفریق)	
5	<< و >>	
6	< و <= و > و >>	
7	== و !=	
8	&	
9	^	
10	|	
11	&&	
12	||	
13	?:	←
14	= و += و -= و *= و /= و %= و &= و ^= و |= و <<= و >>=	←
15	++(پسوندی) و --(پسوندی)	
16	,	

اولویت همهٔ عملگرها در یک نگاه

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ تبدیل انواع در زبان C

- هنگامی که دو نوع مختلف بهمراه هم بکار می روند، رخ می دهد.
- شامل دو نوع است:
  - در عبارت محاسباتی
    - نوع کوچکتر به نوع بزرگتر تبدیل می شود.

```
int  i, j;
char  c;
i = j * c;
```

در عمل c به int تبدیل می شود

  - در احکام انتساب
    - ممکن است بخشی از داده از بین برود.

```
int  i;
chat  c;
i = 259;
c = i;
```

مقدار c برابر 3 خواهد بود! چرا؟

```
int  i;
float  f;
f = 4.6;
i = f;
```

مقدار i برابر ۴ خواهد بود!!!

برنامه نویسی به زبان C

- ✓ تبدیل انواع در زبان C
  - عملیات Type casting با یک مثال مهم:

```
int  i, j, k;
float  f;
i= 4;  j=6;

f = (i + j + 2) / 5;
```

مقدار f برابر 2.0 خواهد بود. اما احتمالا هدف ما چیز دیگری بوده!!

برای رفع این مشکل به صورت زیر عمل می کنیم:

```
f= (float) (i + j + 2) / 5;
```

حال مقدار f برابر 2.4 است.

برنامه نویسی به زبان C

- ✓ انواع خطا در زبان C
  - خطای قاعده ای (Syntax Error) یا خطای زمان کامپایل
    - از قلم افتادن سمی کالن
    - خطاهای تایپی
    - استفاده از یک متغیر بدون تعریف آن
    - و و و
  - خطای زمان اجرا (Run time Error)
    - تقسیم بر صفر (مثال: a = b / c; که در حین اجرا مقدار c برابر صفر تعیین شود!!)
  - خطاهای منطقی (Semantic Error)
    - اشتباه برنامه نویس در پیاده سازی الگوریتم
    - نتیجه: خروجی غیر مطلوب و غیر قابل انتظار (اما در ظاهر بدون خطا)

nastooh@aut.ac.ir

برنامه نویسی به زبان C

- ✓ نکته و عادت مهم در برنامه نویسی
  - ○ جهت افزایش خوانایی برنامه، حتما باید در قسمت های مختلفِ کد، از کامنت استفاده کرد.
    - برای این کار، در ابتدای کامنتها از // استفاده می کنیم.
    - کامنت ها جزو برنامه محسوب نشده و کامپایل نمی شوند!
    - برای ثبت چند خط کامنت آنها را بین /* و */ قرار می دهیم.

```
int   number;  // variable declaration
// Computing Section:
     i = b * 2;
     c = i – b * c;
     w = c / 4;
/*  Coments...
Coments...
Coments...
*/
```


nastooh@aut.ac.ir

برنامه نویسی به زبان C

- تمرین
  - مقدار parts بعد از هر خط چه مقدار می‌باشد؟

```
int parts =5;
parts +=10;
parts -=4;
parts*=7;
```

  - در قطعه برنامه زیر مقدار w برابر است با:

```
int x=2, y=3, z=10;
int w;
w=x*(y-z)+60/x*z;
```

nastooh@aut.ac.ir

برنامه نویسی به زبان C

✓ تمرین

- در قطعه برنامه زیر مقدار w برابر است با:

```
int w,i=12, j=14,k=16;
w=2*i%5*4+j-3/k+2;
```

- در قطعه برنامه زیر مقدار x برابر است با:

```
int x , y = 10;
x=(++y)*(y++);
x*=3;
```

- در قطعه برنامه زیر مقدار w برابر است با:

```
int w , i=6 , j=6 , k=4;
w=i<<2 | j^15 & k>>2;
```

برنامه نویسی به زبان C

- تمرین
  - در قطعه برنامه زیر مقدار x برابر است با:

```
int a=5, b=9, c=15,x;
x=a<<2 | b ^ 7& c>>1;
```

nastooh@aut.ac.ir

پایان

nastooh@aut.ac.ir