

Subject: برنامه‌نویسی

C++

Date: ۰۲/۰۶/۲۳

(Java, PHP, C#, Javascript)

C++

انگیزه (عبدالله)

codeforces, quera, projecteuler.net

geeksforgeeks.org -> آموزش

نام سرگاز:

شماره تماس: ۳ کوئیز - ۵ میانترم - ۵ پایان ترم - ۳ پروژه

اراسته @sharif.edu

محرره ارتباط - شماره

{lynda - coursera - edx}

{resemble.ai} {VAII.E} / {kaggle}

علیون

microsoft (Kinect)

اصل کد در [int main()] Function نوشته شده

```
# ...
#include <iostream>
int main() {
    return 0;
}
```

مانند ساختن خروجی از اول در فنتازیا
به این ها فکر می‌کنیم

std::cout
namespace

فضای استندارد دستورات در اینجا معنی شده اند.

۱) برای استفاده از دستورات و ... راه داریم، ۱) std:: قبل هر دستور وارد کنیم

۲) یا این library ها، قبل int main عبار برای همیشه خط که زیر را وارد کنیم:

using namespace std;

Subject:

Date:

مکثت توسعه مجتمع (IDE) : مجموعه ای که در آن قابلیت های مبتنی بر متن، فرمت کردن، Compile, Debug و ... در آن فراهم آمده .

عنوان در خروجی چاپ می شود

```
cout << "Hello world";
```

نکته ۱: زمین پس برای راحتی قسمت های library - int main() - return 0;

در صورتی که نیاز به توضیح آن نباشد در کدها نوشته نخواهد شد.

```
cout << "Hello" << endl << "world";
```

نکته ۱: زبان C به بزرگی و کوچکی حروف حساس است. Cout ← متغیر به خط دادن

نکته ۱: قابلیت زیاد نویسی برای زبان C مشکل بوجود نمی آورد. (استانده از هر دو راه و فرصت قبل
نکته ۱: آخر هر خط کدها باید و باشد

```
cout << ← دادن خروجی
```

```
cin >> ← گرفتن ورودی
```

سوال ۱: return 0 به چه کار می آید؟

نزدیکی در استفاده آن نیست. این در کدهای طرولان و بزرگی را که از ویرایش پاره شود به دستگاه به ما
عدد غیر صفر را می دهد که می توان با آن وجود خط را فهمید.

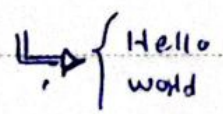
نکته ۱: معمولاً در هر خط در افغان از سطح پایین؛ و استفاده می کنیم که بعد از پایان دستور است.
نکته ۲: طمانی مثل namespace, int, using و... رزور شده اند (این طمانی را طمانی طمانی می نامند) از هسته C و C++ تعریف شده

نکته ۱: می توان با چند روش از coment نویسی استفاده کرد. راه ۱ ← // coment
که از علامت // تا انتهای خط را حریم نوشتن توضیحات قرار می دهد [راه نویسی توضیحات (Single line)]
راه ۲: هر مقدار توضیح بصورتی در چند خط متوالی را در صورت قرار دادن بین * /
می توان در برنامه و میان کردها جاداه

نکته ۳: از مزایای استفاده از روش های فوق برای جد کردن مقادیر از یکدیگر برای Debug کردن
بسیار موثر نام برد.

نکته ۴: C++ نسبت به (مانند C) دارای (لا به بندی) و فاصله های اول خط با دستورات حساس نیست
و می توان حتی چند دستور رو بدون جدییت در ۱ خط نوشت مانند:

```
cout << "Hello" ; cout << "world";
```



نکته ۵: Python نسبت به فاصله اول خط ها حساس است.
float a, b;
cin >> a, b; → Err → cin >> a >> b;

```
10 1/2 → float a, b, x;  
cin >> a >> b; → x = 0.5  
x = b/a;  
cout << "x=" << x;
```

Subject.

Date.

نکته ۱: متغیرها در C++ در زبان‌های برنامه‌نویسی مطلقاً با معنی ریاضی آن‌ها (مثلاً $y = nx$) و به ازای هر x متفاوت است. و ^{میانگین} متغیر فقط یک حافظه یا ظرف است.
به مثال زیر توجه کنید:

```

10 | float a, b, x;
5  | cin >> a >> b;
   | x = -b/a;
   | a = 5;
   | cout << x;

```

-> -0.5

نکته ۲: دستورات حافظه کامپیوترها (در نهایت) یا نامحدود نیست. بنابراین وقتی دستور `cout << 2/3` را بزنیم، جواب قابل عدد با $2/3$ یکسان نیست. تعداد رقم اعشار محدود است.
فوق نمی‌گذرد متغیر با `Enter` یا `Space` در ورودی از هم جدا شده باشد.

نکته ۳: در زبان C++ هیچ خط اولی که `float a=0, b=0, x=0;` بنویسیم هیچ نظریه ندارد. نویسیم با تغییر در عدد برنامه به ما هشدار داده است. ما متغیر را مقدار دهی کردیم و استفاده کردیم و مقدار را دوباره بعد از تعیین کردیم. (فارم بهبوده کرده اند)

نکته ۴: مقدار دهی اولیه الزامی است. برنامه انجام شود به صورت تصادفی خواهد بود.

نکته ۵: اگر یک متغیر یا عدد در این اعشار را در متغیر عدد صحیح بنویسیم با اعشار بریده خواهد شد.

```

int a, b; float x;
3  | cin >> a >> b;
8  | x = -b/a;
   | cout << "x=" << x << endl;

```

-> -2

نتیجه گرفتن را تقدم اعمال برهم حاصل می‌شود. که ما اول از حافظه b را حساب کرده سپس $-b/a$.

رابطه توابع به نوع در حافظه بصورت int ذخیره کرده پس حاصل را به مضامین حافظه متغیر x انتقال می دهد

راه عمل: جای خط سوم که دوباره را می بینیم $x = -\text{Float}(a)/a$

البته در راه فوق دیگر لزومی نیست حتما هر دو متغیر a و b را در Float بنویسیم

نکته: این خط که موقعا فرمت متغیر b را Float در نظر می گیرد.

عالمی $\frac{x^2 - y^2}{a/b} \Rightarrow (x*x - y*y) / (a*b/c)$

اولیهاها، توان؟ برانتر ← ضرب و تقسیم ← جمع و تفریق ← باقی مانده؟ در جاهای هم اولیها، اول نسبت چپ انجام می شود

```
int a=-10, b=7;
```

```
cout << (a/b) << " " << (a%b);  $\Rightarrow -1, -3$ 
```

نکته: علامت باقی مانده با علامت عدد اول یکی است. (مفروضه + و - آن است)

تقسیم و باقی مانده
~~int x=8, y=10;~~

```
int x=4, y=8, z=10;
```

```
x++;
```

```
++x;
```

```
y--;
```

```
cout << x << " " << y << endl;  $\Rightarrow$  5 7
```

```
z = x++;
```

```
cout << z << " " << x << endl;  $\Rightarrow$  5 9
```

```
z = 10;
```

```
x = 6;
```

```
z = ++x;
```

```
cout << z << " " << x << endl;  $\Rightarrow$  7 7
```

Subject.

Date.

```

int x = 5, y = 9, z = 7;
x = x + 2;
y = y - 4;
z = z * 3;
}
int x = 0, y = 9, z = 7;
x = 2;
y = 5;
z = 21;
cout << x << " " << y << " " << z; // 2 5 21

```

در حالت طی من توان نوشت:

$x \square = y$ برابر است $x = x \square y$

نکته: \square می تواند یک عمل باشد!

```
#define Max 100
```

```
#define sum(a,b) a+b
```

```

int main() {
    int x, y, z;
    cin >> x >> y;
    cout << Max * x << endl;
    cout << x << "+" << y << "=" << sum(x,y) << endl;
    z = sum(x,y) * sum(x,y) // z = x+y * x+y
    cout << z << endl;
}

```

2 3

```

200
2+3=5
11

```



ورودی

خروجی