

کارگاه کامپیوتر

جلسه : سوم

مدرس : الهام یوسفی





آشنایی با زبان C++

تعریف ثابت

تبدیل نوع

توابع کتابخانه‌ای

تعریف ثابت

مقادیر متغیرها می‌توانند در طول اجرا تغییر کنند. اما گاهی لازم است مقادیری داشته باشیم که در طول برنامه تغییر نکنند مانند عدد p در محاسبات ریاضی. در اینصورت ثابت‌ها بصورت زیر باید تعریف شوند :

define مقدار نام متغیر

const نوع مقدار = نام متغیر

- در استفاده از پیش پردازنده `define` عملاً متغیری تعریف نمی‌شود بلکه فقط یک جای‌گذاری ساده انجام می‌شود.
- در `const` یک متغیر است که هم آدرس دارد و هم دارای همه ویژگی‌های متغیر با این تفاوت که ایستا است.

مثال :

```
# define    P    3.14
```

```
const float    Pi = 3.14;
```

تبدیل نوع (CASTING)

▪ وقتی یک نوع متغیر را به نوع دیگر تبدیل کنیم آن را Cast کرده ایم.

▪ انواع تبدیل صریح :

- **(float) i** : تبدیل مقدار i به عدد اعشاری
- **(int) f** : تبدیل مقدار اعشار f به عدد صحیح
- **(int) c** : تبدیل مقدار کاراکتر c به کد اسکی (که یک عدد صحیح است)
- **(char) a** : تبدیل مقدار صحیح a به کاراکتر معادل

مثال :

```
cout << "10/3 = " << 10/3 ;    // 3
cout << "\nChange to float 10/3 = " << (float)10/3 ;    // 3.33333
cout << "\nChange to int 3.1415 = " << (int)3.1415 ;    // 3
cout << "\nChange to ASCII character A = " << (int)'A' ;    // 65
cout << "\nChange to character = " << (char) 65 ;    // A
```

مثال

• یک برنامه ای بنویسید که : (فایل Class3_Casting.cpp)

• عدد اعشاری را بصورت ثابت تعریف کرده سپس در خروجی مقدار صحیح آن چاپ شود.

• سپس دو عدد صحیح دریافت کند و حاصل تقسیم آنها را بصورت اعشار چاپ کند.

• عددی دریافت کند کاراکتر معادل آن را نمایش دهد.

• کاراکتری دریافت نماید و کد اسکی آن را چاپ کند.

```
#include <iostream>
using namespace std;
#define pi 3.14159
const float p=3.14159;
main()
{
```

```
    int a,x,y;
    char b;
    cout <<"Define Pi = " <<pi <<"\nConstant Pi = " <<p;
    cout <<"\nChange to int Number = " << (int)p;
    //p++; //Error
```

پیام خطا می دهد زیرا p ثابت است و نمی توان به آن یکی اضافه کرد.

```
    cout << "\n\nEnter x,y : ";
    cin >> x >> y;
    cout <<"x/y = " << x/y <<"\nChange to float : " << (float)x/y;
```

```
    cout << "\n\nEnter a number : ";
    cin >> a;
    cout <<"Character = " <<(char)a;
```

```
    cout << "\n\nEnter a character : ";
    cin >> b;
    cout <<"ASCII code = " <<(int)b;
```

```
}
```

```
Define Pi = 3.14159
Constant Pi = 3.14159
Change to int Number = 3

Enter x,y : 7
2
x/y = 3
Change to float : 3.5

Enter a number : 97
Character = a

Enter a character : A
ASCII code = 65
```

توابع

▪ دو نوع تابع وجود دارد :

- توابع کتابخانه‌ای
- توابع نوشته شده توسط کاربر

□ توابع کتابخانه‌ای

- این توابع از پیش ساخته شده هستند.
- برنامه نویس می‌تواند با فراخوانی توابع کتابخانه‌ای از آن‌ها استفاده کند و احتیاجی به پیاده‌سازی نیست.

نام تابع	توضیح	فایل سرآیند
<code>sqrt(x)</code>	جذر x	<code>math.h</code>
<code>pow(x,p)</code>	x به توان p	<code>math.h</code>
<code>abs(x)</code>	قدر مطلق x	<code>math.h</code>
<code>setw(i)</code>	ایجاد i کاراکتر فاصله	<code>iomanip</code>
<code>setprecision(i)</code>	نمایش عدد اعشار با دقت i (با در نظر گرفتن خود اعشار) در صورت نیاز عدد گرد می‌شود.	<code>iomanip</code>

- استفاده از تابع `setprecision()` برای نمایش عدد اعشاری: (فایل `Class3_setpreciosion.cpp`)
- دقت 0 و 1 در تابع هیچ اعشاری نمایش نمی‌دهد اما در قسمت صحیح عدد گرد شده نمایش داده می‌شود.
- در تمام موارد استفاده از تابع، با توجه به قسمت اعشاری عدد، در خروجی عدد گرد شده است. (حتماً اعشار خواهد داشت)
- اگر دقت اعشار منفی در نظر گرفته شود، در نمایش اعشار عدد تاثیری نخواهد داشت.

```
#include <iostream>
#include <iomanip>
using namespace std;
const float x=1.87654;
main()
{
    cout <<x;
    cout <<endl<<setprecision(0) <<x;
    cout <<endl<<setprecision(1) <<x;
    cout <<endl<<setprecision(2) <<x;
    cout <<endl<<setprecision(3) <<x;
    cout <<endl<<setprecision(4) <<x;
    cout <<endl<<setprecision(5) <<x;
    cout <<endl<<setprecision(-1) <<x;
    cout <<endl<<setprecision(-2) <<x;
}
```

```
1.87654
2
2
1.9
1.88
1.877
1.8765
1.87654
1.87654
```

- برنامه ای بنویسید که دو عدد از ورودی دریافت کند سپس از توابع کتابخانه ای صفحه قبل استفاده نمایید. (فایل Class3_Function_Lib.cpp)

۱- بعد از پیام مورد نظر، X به توان Y نمایش داده می شود.

۲- در خط جدید بعد از نمایش پیام، جذر X نمایش داده می شود سپس یک فاصله ۳۰

کاراکتری ایجاد می شود و بعد از پیام، جذر Y نمایش داده می شود.

۳- مانند دستور قبل است با این تفاوت که دقت اعشار مشخص شده است.

۴- به جای دستور بالا می توان از دو دستور Comment شده استفاده کرد. یک بار از تابع setprecision استفاده

کرد و تمام دستورات بعدی اعشار با دقت مورد نظر نمایش داده شود.

۵- ابتدا منفی متغیر X نمایش داده می شود سپس قدر مطلق منفی X.

```
#include <iostream>
#include <math.h>
#include <iomanip>

using namespace std;
main()
{
    int x,y;
    cout << "Enter x , y =";
    cin >> x>>y;
    cout << "Pow(x,y) = " << pow(x,y) ; //۱
    cout <<"\nSqrt(x) = " << sqrt(x) <<setw(30) <<"Sqrt(y) = " <<sqrt(y); // ۲
    cout <<"\nSqrt(x) = " << setprecision(3) <<sqrt(x) <<setw(30) <<"Sqrt(y) = " <<setprecision(3) <<sqrt(y); // ۳
    /*cout << setprecision(3) ;
    cout <<"\nSqrt(x) = " <<sqrt(x) <<setw(30) <<"Sqrt(y) = " <<sqrt(y);*/ } ۴
    cout <<"\n-x = " << -x << " \t" << "|-x| = " << abs(-x); //۵
}
```

```
Enter x , y =2
4
Pow(x,y) = 16
Sqrt(x) = 1.41421          Sqrt(y) = 2
Sqrt(x) = 1.41           Sqrt(y) = 2
-x = -2                |-x| = 2
```


تمرین

۱- برنامه ای بنویسید که عدد صحیح a را دریافت کند و عبارت زیر را با استفاده از توابع محاسبه نماید. (جواب بصورت اعشار با ۳ رقم اعشار)

$$\frac{a^3}{a^2 + a + 1}$$

۲- برنامه ای بنویسید که دو عدد از ورودی دریافت کرده و حاصلضرب قدرمطلق آنها را در خروجی چاپ کند.

۳- برنامه ای بنویسید که شعاع دایره را دریافت کند سپس مساحت و محیط دایره را با فاصله ۵۰ کاراکتر از هم چاپ نماید. (عدد P بصورت ثابت تعریف شود)

۴- برنامه ای بنویسید که یک کاراکتر دریافت کرده سپس کد اسکی آن را نمایش دهد. همچنین یک عدد دریافت کند و کاراکتر معادل آن را چاپ کند.