# پیش نیاز برنامه نویسی

بخش عمده‌ای از دوره آموزشی میکرودمی را برنامه نویسی تشکیل می‌دهد. به همین دلیل تصمیم گرفتیم بخشی را به مبانی برنامه نویسی اختصاص بدهیم تا بتوانیم بهتر و دقیق‌تر به برنامه نویسی بپردازیم. در این بخش به بررسی دستورهای اصلی که در حین دوره به کار ما می‌آید می‌پردازیم.

# عملگرهای برنامه نویسی

برنامه نویسی چیزی جز منطق و ریاضیاتی که ما بر روی برگه از آنها استفاده می کنیم نیست. در ریاضیات نماد‌های =، +، -، <، >، \* و / معنا و مفهوم خاصی دارند که در برنامه نویسی نیز همان معانی را دارند، با این تفاوت که ممکن است نگارش آنها متفاوت باشد. در ادامه به بررسی هر یک می‌پردازیم.

## + ، -

این عملگر‌ها در برنامه نویسی عملیات جمع و تفریق را انجام می‌دهند.

=

این عملگر در برنامه نویسی معنی متفاوتی دارد. در برنامه نویسی متغیری که در سمت چپ این عملگر است متغیری تعریف شده می‌باشد که قرار است مقدار متغیری را که در سمت راست مساوی هست به خاطر بسپارد. در واقع هر آنچه که در سمت راست این عملگر هست به داخل هر آنچه که در سمت چپ این عملگر است ریخته می‌شود. به مثال‌های زیر توجه کنید:

کد 1 :



گفتنی است که می‌توان بر روی یک متغیر در سمت راست عملیاتی انجام داد و در داخل همان متغیر ذخیره کرد.

کد 2:



## \* ، / ، %

دو عملگر اول در برنامه نویسی عملیات ضرب و تقسیم را انجام می‌دهنددر صورتی که عملگر "%" در برنامه‌نویسی باقی مانده یک تقسیم را خروجی می‌دهد.

کد 3:



تا اینجا عملگر‌هایی را که مربوط به عملیات‌های ریاضی بودند بررسی کردیم. در ادامه به بررسی عملگر‌های مربوط به مقایسه‌ی ریاضیاتی می‌پردازیم.

< == >

این عملگرها به ترتیب وضعیت کوچکتر، مساوی بودن و بزرگتر بودن بودن سمت راست را نسبت به سمت چپ بررسی می‌کنند. همین طور <= و >= به ترتیب نماد کوچکتر مساوی و بزرگتر مساوی‌ هستند. خروجی این عملگر‌ها درستی این وضعیت است.

کد 4:



# عملگر های شرطی

if

این یک عملگر شرطی است. همواره به یاد داشته باشید که در فضای برنامه نویسی C++ علامت ";" تمام شدن گزاره را نشان می دهد. نحوه‌ی استفاده از عملگر شرطی if در فضای برنامه نویسی آردینو به صورت زیر است:

کد 5:



## if else

در صورت وجود فقط یک شرط می توانیم از if استفاده کنیم. اگر تعداد شرط‌ها بیشتر بود، اولین شرط را با if بیان می کنیم و بقیه شروط را با if else بیان می کنیم. به مثال زیر توجه کنید:

کد 6:



## else

اگر تمامی شروط بیان شده برقرار نباشد، شرط else برقرار می‌باشد. در واقع else به معنی این است که اگر هیچ یک از شروط بیان شده بر قرار نشد، عملیات بیان شده داخل else انجام شود. به مثال زیر توجه کنید:

کد 7:



توجه شود که در صورت وجود چندین if وif else ، else به نزدیک ترین if قبل از خود اشاره می‌کنند. به مثال زیر توجه کنید:

کد 8:



# عملگر های حلقه‌ای

## while

این عملگر تا وقتی شرطش برقرار باشد عملیاتش را انجام می‌دهد. به بیان دیگر این عملگر تا زمانی که شرطش درست باشد، حلقه‌ای از کد را تکرار می‌کند. به مثال زیر توجه کنید:

کد 9:



چون مقدار a همواره یک عدد مثبت است و صفر نیست، پس عملیات داخل عملگر تا بینهایت اجرا خواهد شد. در اینجا شما یک حلقه‌ی بینهایت ساخته‌اید و نرم افزار در اینجا گیر خواهد کرد. برای جلوگیری از چنین اتفاقی شما می‌توانید از دستور break استفاده کنید. این دستور برنامه را از داخلی ترین حلقه بیرون خواهد آورد. به مثال زیر توجه کنید:

کد 10:



البته می توانید به جای استفاده از دستور break از شرط while استفاده کنید:

کد 11:



## for :

این عملگر نیز مانند عملگر while است با این تفاوت که نه تنها شرط دارد، بلکه شمارنده نیز دارد. در واقع بعد از هر بار تمام شدن حلقه، شمارنده مقدارش کم یا زیاد می‌شود و بعد شرط بررسی می‌شود. در صورت برقرار بودن شرط، حلقه اجرا می‌شود. به مثال زیر توجه کنید:

کد 12:



توجه: نماد i++ یعنی مقدار i با عدد یک جمع شود. تمامی کد های 10، 11 و 12 همگی 9 بار اجرا می‌شوند و با یکدیگر فرقی ندارند.

# متغیرها

این مبحث در بخش متغیر ها به صورت کامل بحث خواهد شد و در اینجا فقط بخشی از آن که در طول برنامه‌های ابتدایی به آن ها نیاز داریم گفته خواهد شد. در اینجا بیشتر نحوه تعریف یک متغیر مد نظر است. متغیر در واقع فضایی از حافظه است که در خود داده‌ای را که ساختار آن از قبل مشخص شده است ذخیره می‌کند. مانند متغیر های جبری که در جبر همگی با آنها آشنا شده‌ایم. ساختار این داده ها توسط معرف این داده مشخص می‌شود. به طور مثال ساختار اعداد صحیح و اعداد طبیعی با یک دیگر فرق دارد پس نیازمند معرف‌های مجزا هستند. هر نوع داده در فضای برنامه نویسی C++ تعریف شده است و نام بخصوصی دارد. به طور مثال اعداد صحیح با نماد int تعریف می‌شوند. همان طور که گفته شد هر نوع داده فضایی را در حافظه ذخیره می‌کند. به طور مثال اعداد صحیح 2 بایت از فضا را ذخیره می‌کنند. به همین دلیل اگر متغیری صحیح (int) تعریف شود، فقط می‌تواند اعداد بین 32768 تا 32768- را در خود ذخیره کند. حال اگر نیازمند ذخیره‌سازی عددی بزرگتر بودیم باید از نوع دیگری از داده استفاده کنیم (به طور مثال از long).

کد 13:

