

فصل ۱: آشنایی با مفاهیم پایداری پردازش داده‌ها

تعریف داده:

مجموعه‌ای از مقادیر در مورد یک موضوع یا شیء است که به صورت کمی یا کیفی نمایش داده می‌شود، مثل نمرات ریاضی و اسامی دانش‌آموزان یک کلاس و یا اندازه‌ی دمای محیط.

تعریف اطلاعات:

نتایج حاصل از انجام محاسبات بر روی داده‌ها را اطلاعات می‌نامند. مثل معدل نمرات ریاضی یک کلاس.

تعریف دانش:

نتیجه‌ی تفسیر و بررسی اطلاعات را دانش می‌نامند که می‌تواند مبنای تصمیم‌گیری‌ها باشد.

تعریف پردازش:

مجموعه محاسبات و عملیاتی است که روی داده‌ها صورت می‌گیرد. مثل محاسبه‌ی مجموع یا خارج قسمت دو عدد یا تشخیص شماره پلاک خودرو از روی عکس.

نکات مهم در فرآیند به‌دست آوردن اطلاعات:

- ۱- صحت داده‌ها یعنی انتخاب داده‌های درست برای پردازش.
- ۲- درستی انجام محاسبات یعنی انجام عملیات با دقت کامل.
- ۳- روش انجام پردازش یعنی انجام عملیات محاسباتی مناسب روی داده‌ها.
رشته‌ها و توابع رشته‌ای

تعریف برنامه:

به مجموعه دستوراتی که تعیین می‌کند کامپیوتر چه نوع پردازشی را روی داده‌ها انجام دهد و اطلاعات به‌دست آمده را چگونه نمایش دهد، برنامه می‌گویند.

تعریف زبان ماشین:

به زبان قابل فهم برای کامپیوتر که دنباله‌ای از کدهای صفر و یک است زبان ماشین می‌گویند.

تعریف برنامه‌نویسی:

نوشتن دستورات لازم برای کنترل نحوه‌ی کار کامپیوتر را برنامه‌نویسی می‌گویند.

تعریف مترجم یا کامپایلر:

یک برنامه‌ی کامپیوتری است که وظیفه‌ی آن ترجمه‌ی برنامه نوشته شده به یک زبان، به کدهای زبان ماشین است.

انواع زبان‌های برنامه‌نویسی:

- ۱- زبان‌های سطح پایین: به زبان‌هایی گفته می‌شود که به زبان پردازنده‌ی کامپیوتر (CPU) نزدیک است، مثل زبان ماشین و زبان اسمبلی.
 - ۲- زبان‌های سطح بالا: به زبان محاوره‌ای انسان نزدیک است، مثل زبان VB، JAVA و C#.
 - ۳- زبان‌های سطح میانی: دستورات آن‌ها برای دسترسی راحت‌تر به سخت‌افزار پیش‌بینی شده و نیز به زبان عامیانه‌ی ما نزدیک است مثل زبان C.
- ◆ نکته: از زبان‌های سطح میانی برای کار با سخت‌افزار کامپیوتر و برنامه‌ریزی وسایلی که دارای پردازنده می‌باشند استفاده می‌شود.

تست‌های فصل اول

- ۱- کدام گزینه جزو زبان‌های سطح میانی می‌باشد؟
 C# (۱) C (۲) VB (۳) Java (۴)
- ۲- تمام جملات زیر صحیح می‌باشند به جز:
 (۱) کامپیوتر پردازشگر داده‌ها است.
 (۲) زبان C# جزو زبان‌های سطح بالا می‌باشد.
 (۳) نتایج بررسی اطلاعات به دانش ختم می‌شود.
 (۴) نتایج تفسیر داده‌ها به دانش ختم می‌گردد.
- ۳- تعریف زیر مربوط به کدام زبان برنامه‌نویسی می‌باشد؟
 «زبان قابل فهم برای کامپیوتر که به صورت دنباله‌ای از صفر و یک‌ها است.»
 (۱) زبان اسمبلی
 (۲) زبان سطح بالا
 (۳) زبان ماشین
 (۴) زبان سطح میانی
- ۴- کدام گزینه در فرآیند رسیدن از داده‌ها به اطلاعات ضروری نیست؟
 (۱) نتایج حاصل از انجام محاسبات
 (۲) درستی انجام محاسبات
 (۳) صحت داده‌ها
 (۴) روش انجام پردازش
- ۵- کدام گزینه صحیح است؟
 (۱) وظیفه‌ی مترجم اجرای برنامه‌ها می‌باشد.
 (۲) زبان قابل فهم کامپیوتر زبان سطح میانی است.
 (۳) اطلاعات می‌تواند مبنای تصمیم‌گیری‌ها باشد.
 (۴) زبان‌های C و C# هر دو جزو زبان‌های سطح میانی می‌باشند.
- ۶- تمام موارد زیر جزو زبان‌های سطح بالا می‌باشند، به جز:
 Java (۱) C (۲) C# (۳) VB (۴)
- ۷- کدام یک از جملات زیر صحیح است؟
 (۱) کامپایلر سخت‌افزاری است که وظیفه‌ی آن ترجمه‌ی برنامه به کدهای زبان ماشین است.
 (۲) داده‌ها مجموعه‌ای از مقادیر در مورد یک موضوع یا شیء می‌باشد.
 (۳) داده‌ها پس از پردازش تبدیل به دانش می‌شود.
 (۴) از زبان‌های سطح میانی برای کار با سخت‌افزار کامپیوتر استفاده می‌شود.
- ۸- مجموعه‌ای از مقادیر در مورد یک موضوع که به صورت کمی یا کیفی نمایش داده می‌شود، چه نام دارد؟
 (۱) داده‌ها (۲) اطلاعات (۳) پردازش (۴) خروجی
- ۹- به نتایج حاصل از انجام محاسبات بر روی داده‌ها چه می‌گویند؟
 (۱) پردازش (۲) اطلاعات (۳) دانش (۴) برنامه
- ۱۰- تعریف زیر مربوط به کدام گزینه است؟
 «مجموعه دستوراتی که تعیین می‌کند کامپیوتر چه نوع پردازشی را روی داده‌ها انجام دهد.»
 (۱) پردازش (۲) Program (۳) Process (۴) Compiler

فصل ۲ : آشنایی با زبان C#

زبان C# یک زبان سطح بالا، شی‌گرا و همه منظوره و شبیه C++ و Java می‌باشد که توسط مایکروسافت ابداع شده است. زبان C که سیستم عامل Unix به وسیله‌ی آن نوشته شده یک زبان حرفه‌ای با انعطاف بالا است که در آن امکان دسترسی به سخت‌افزار برای برنامه‌نویس وجود دارد و اشتباهات منطقی برنامه‌نویس را کم‌تر کنترل می‌نماید.

ولی C# در هنگام ترجمه و اجرای برنامه عمل تطبیق و به‌کارگیری داده‌ها را به دقت انجام می‌دهد تا از اشتباهات دستوری جلوگیری نماید. نرم‌افزارهای اداری، برنامه‌های تحت وب، نرم‌افزارهای تلفن همراه و بازی‌های کامپیوتری با زبان C# و با استفاده از لایه NET، تولید می‌شود.

شروع برنامه‌نویسی

مراحل نوشتن، ترجمه و اجرای برنامه به شکل زیر است:

۱- نوشتن کد برنامه در یک ویرایشگر ساده مثل Notepad و ذخیره‌ی آن با پسوند CS.

۲- ترجمه‌ی برنامه به وسیله‌ی مترجم زبان C# به نام CSC.EXE

۳- اجرای برنامه‌ی ترجمه شده

اولین برنامه به زبان C#

ابتدا به برنامه‌ی ساده‌ی زیر توجه نموده، سپس با واژه‌ها و کلمات به‌کار رفته در آن آشنا می‌شوید:

```
class WelcomeToC
{
static void Main ()
{
System.Console.WriteLine("Welcome to C# ");
//sert a blank line
System.Console.WriteLine();
System.Console.WriteLine("This is my first Program");
}
}
```

◆ نکته مهم: زبان C# مثل زبان C، C++ و Java نسبت به حروف کوچک و بزرگ حساس است. کلماتی مثل static و void باید با حروف کوچک نوشته شود و حرف اول کلمات Console، System، Main و WriteLine باید حروف بزرگ باشد. برنامه‌ی فوق در سطر اول پیام Welcome to C#، سپس یک سطر خالی و در سطر سوم عبارت This is my first program را نمایش می‌دهد.

تعریف کلاس (class):

قالب یا الگویی است که در آن داده‌هایی تعریف می‌شود که مربوط به یک موضوع است و می‌توان عملیاتی بر روی داده‌ها انجام داد. مثلاً Console یک کلاس آماده در C# است که عملیات مختلف ورودی و خروجی (صفحه‌کلید و صفحه نمایش) در آن تعریف شده است.

تعریف متد:

مجموعه‌ای از دستورات است که برای انجام یک کار لازم است. هر متد مطابق با عملکردش نام‌گذاری می‌شود و دارای () است که ممکن است ورودی‌هایی در آن ذکر شود.

کلمات کلیدی یا (رزرو شده): کلمات شناخته شده‌ای هستند که دارای معنی و مفهوم مشخصی می‌باشند مثل static و class و void.

تعریف شناسه:

نامی است که به وسیله‌ی برنامه‌نویس انتخاب می‌شود.

متد () Main:

بدنه‌ی اجرایی برنامه است و هر دستوری که در این قسمت نوشته شود به وسیله‌ی کامپیوتر به ترتیب اجرا می‌شود. به عبارتی متد () Main نقطه‌ی آغاز اجرای برنامه است و اجرای یک برنامه از اولین دستور داخل آن شروع می‌شود.

کلمات static و void ویژگی‌های متد را مشخص می‌کنند.

◆ کانون فرهنگی آموزش ◆

درج توضیحات: برای درج توضیحات در برنامه اگر یک خط باشد از علامت // و اگر چند خط باشد از علامت /* / استفاده می‌شود. توضیحات به وسیله کامپایلر ترجمه و اجرا نمی‌شود.

برای درج یک سطر خالی روی صفحه‌ی نمایش از متد WriteLine() به شکل زیر استفاده می‌شود.

`Console.WriteLine()`

◆ نکته: برای کوتاه کردن دستورات می‌توانیم در ابتدای برنامه فضای نامی System را معرفی کنیم که در آن کلاس Console تعریف شده است و در این صورت می‌توانیم در داخل برنامه، کلمه‌ی System را از ابتدای دستورات حذف کنیم.

`using System;`

بدین منظور در ابتدای برنامه دستور مقابل را اضافه می‌کنیم.

◆ نکته: در زبان C# انتهای هر دستور با علامت ; مشخص می‌شود.

متد () ReadKey:

با اجرای این متد، پس از ترجمه و اجرای برنامه، کامپیوتر منتظر زدن یک کلید توسط کاربر باقی می‌ماند و بدین ترتیب کاربر می‌تواند صفحه‌ی کنسول یا خروجی برنامه را تا زمان دلخواه مشاهده نماید.

کار با رنگ‌ها:

ویژگی BackgroundColor رنگ زمینه‌ی کنسول و ویژگی ForegroundColor رنگ نوشته یا قلم کنسول را تعیین می‌کند.

مثال: می‌خواهیم کلمه IRAN را به رنگ قرمز روی زمینه‌ی سبز مشاهده نماییم.

```
Console.BackgroundColor = ConsoleColor.Green;
Console.ForegroundColor = ConsoleColor.Red;
Console.WriteLine("IRAN");
```

◆ نکته: برای پاک کردن کنسول یا صفحه‌ی خروجی از متد Clear به شکل زیر استفاده می‌شود:

`Console.Clear();`

◆ نکته: برای ترجمه‌ی برنامه از کامپایلر CSC.exe استفاده می‌شود که در برنامه‌ی Net Framework قرار دارد. پس از ترجمه‌ی برنامه، فایل جدیدی با پسوند EXE ساخته می‌شود که برای اجرای برنامه قابل استفاده است.

تست‌های فصل دوم

- ۱- سیستم عامل Unix به وسیله‌ی کدام زبان نوشته شده است؟
 (۱) Java (۲) C (۳) C# (۴) C++
- ۲- زبان C# یک زبان سطح ... و شبیه زبان ... می‌باشد.
 (۱) میانی - C++ (۲) میانی - Java (۳) بالا - C++ (۴) بالا - Pascal
- ۳- مجموعه‌ای از دستورات که برای انجام یک کار لازم است را چه می‌نامند؟
 (۱) کلاس (۲) شناسه (۳) روال (۴) متد
- ۴- نقطه‌ی آغاز اجرای برنامه کدام است؟
 (۱) متد () Main (۲) کلاس Console (۳) using (۴) System
- ۵- برای درج یک سطر خالی روی صفحه نمایش از کدام دستور استفاده می‌شود؟
 (۱) Console.WriteLine; (۲) Console.Write; (۳) Console.WriteLine (); (۴) Console.Write ();
- ۶- در زبان C# انتهای هر دستور با کدام علامت مشخص می‌شود؟
 (۱) ; (۲) ; (۳) // (۴) /*
- ۷- برای پاک کردن کنسول یا صفحه‌ی خروجی از کدام متد استفاده می‌شود؟
 (۱) Cls (۲) Cls() (۳) Clear (۴) Clear ()
- ۸- برای تغییر رنگ قلم به رنگ قرمز کدام دستور صحیح است؟
 (۱) Console.BackgroundColor = ConsoleColor.Red (۲) Console.ForegroundColor = ConsoleColor.Red (۳) Console.Background = ConsoleColor.Red (۴) Console.Foreground = ConsoleColor.Red
- ۹- برای درج توضیحات یک خطی از کدام گزینه در ابتدای سطر استفاده می‌شود؟
 (۱) /* (۲) /* (۳) // (۴) //
- ۲۰- تمام جملات زیر صحیح می‌باشند به جز:
 (۱) زبان C# نسبت به حروف کوچک و بزرگ حساس است.
 (۲) اگر در ابتدای برنامه کلمه‌ی System را اضافه کنیم، می‌توانیم در داخل برنامه کلمه‌ی System را از ابتدای دستورات حذف کنیم.
 (۳) Console یک کلاس آماده در C# است.
 (۴) کلاس، قالب یا الگویی است که در آن داده‌هایی تعریف می‌شود.

فصل ۳: آشنایی با ویژوال استودیو

محیط IDE، به محیط برنامه‌نویسی گفته می‌شود که در آن می‌توان تمام مراحل برنامه‌نویسی، ترجمه، اشکال‌یابی و اجرای برنامه را انجام داد و به معنای محیط تولید برنامه‌ی متمرکز می‌باشد. ویژوال استودیو محیط IDE شرکت مایکروسافت است که بسیار پیشرفته است و از چند زبان C#، C++ و VB پشتیبانی می‌کند. با توجه به این‌که تایپ برنامه در یک ویرایشگر، ورود به پنجره‌ی فرمان، ترجمه کردن، عیب‌یابی و اشکال‌زدایی برنامه، عملیاتی وقت‌گیر و پر زحمت می‌باشد، در محیط ویژوال استودیو همه‌ی ابزارها و لوازم مورد نیاز برنامه‌نویسی گردآوری شده است.

ایجاد یک پروژه جدید

با اجرای ویژوال استودیو در صفحه‌ی Start Page گزینه‌ی New Project را انتخاب و در کادر محاوره‌ی New Project از پانل سمت چپ Visual C# و از پانل وسط گزینه‌ی Console Application را انتخاب می‌کنیم. در بخش Name نام پروژه و در قسمت Location مسیر ذخیره‌ی پروژه و در قسمت Solution نام پوشه‌ای که فایل‌های مربوط به یک یا چند پروژه در آن نگهداری می‌شود را تعیین می‌کنیم. در پنجره‌ی Error List، خطاها و اشکالات تایپی یا ساختاری برنامه لیست می‌شوند. در پنجره‌ی Solution Explorer ساختار پروژه و تمام فایل‌های موجود در آن نشان داده می‌شود.

دستور namespace

با دستور namespace یک فضای نامی جدید تعریف می‌شود که برای سازماندهی و دسته‌بندی پروژه‌های بزرگ مورد استفاده قرار می‌گیرد.

◆ نکته: برای ترجمه‌ی برنامه از کلید F6 استفاده می‌شود.

◆ نکته: برای اجرای برنامه از دکمه‌ی ▶ در نوار ابزار و یا کلید ترکیبی Ctrl + F5 استفاده می‌شود.

متد () Beep

متد () Beep از کلاس کنسول برای ایجاد یک صوت استفاده می‌شود و شکل کلی آن به صورت زیر است:

Console.Beep(); (فرکانس بر حسب هرتز، مدت زمان بر حسب میلی‌ثانیه)

البته متد () Beep را بدون پارامتر نیز می‌توان به کار برد ولی برای تغییر مدت زمان نواختن صدا و تغییر فرکانس صدا می‌توانیم مقادیر مورد نظر را در داخل پراتنز بنویسیم.

◆ نکته: محدوده‌ی شنوایی گوش انسان فرکانس ۲۰۰ تا ۱۰۰۰۰ هرتز می‌باشد.

تست‌های فصل سوم

۲۱- ویژوال استودیو از کدام زبان پشتیبانی می‌کند؟ C# (۱) C++ (۲)	۳ Visual Basic	۴ هر سه مورد
۲۲- پوشه‌ای که فایل‌های پروژه در آن نگهداری می‌شود چه نام دارد؟ Solution Explorer (۲) Solution name (۱)	۳ namespace	۴ Solution Folder
۲۳- برای ترجمه‌ی برنامه از کدام کلید استفاده می‌شود؟ F5 (۱) Ctrl + F5 (۲)	۳ F6	۴ F4
۲۴- برای ایجاد یک صوت از کدام متد استفاده می‌شود؟ Sound () (۱) Beep () (۲)	۳ Voice ()	۴ Console.Beep
۲۵- در محیط ویژوال استودیو علاوه بر تایپ برنامه، می‌توان برنامه را ... ، عیب‌یابی و سرانجام ... کرد. (از راست به چپ) (۱) ترجمه - اجرا (۲) اجرا - ترجمه (۳) خطایابی - ترجمه (۴) اشکال‌زدایی - اجرا	۳ static void	۴ exe
۲۶- IDE مخفف کدام گزینه است؟ Interface Display Explorer (۱) Integrated Development Environment (۳)	۳ using	۴ CS
۲۷- عملکرد دستور Console.Beep(1000, 2000) چیست؟ (۱) نواختن صدا به مدت یک ثانیه با فرکانس ۲۰۰۰ هرتز (۲) نواختن صدا به مدت دو ثانیه با فرکانس ۱۰۰۰ هرتز (۳) نواختن صدا به مدت هزار ثانیه با فرکانس ۲۰۰۰ هرتز (۴) نواختن صدا به مدت ۲۰۰۰ ثانیه با فرکانس ۱۰۰۰ هرتز	۳ Ctrl + F5	۴ Ctrl + F5 یا F5
۲۸- برای اجرای برنامه از کدام کلید استفاده می‌شود؟ F5 (۱) F6 (۲)	۳ class	۴ namespace
۲۹- از کدام دستور برای تعریف یک فضای نامی جدید استفاده می‌شود؟ namespace (۲) class (۱)	۳ C	۴ CSC
۳۰- فایل برنامه با چه پسوندی ذخیره می‌شود؟ C (۲) CSC (۱)		

فصل ۴: آشنایی با انواع داده و متغیرها

تعریف متغیر:

متغیر مکانی از حافظه RAM است که برای نگهداری موقتی داده‌ها یا اطلاعات استفاده می‌شود و مقدار آن در طول برنامه قابل تغییر است. مقدار جدید جایگزین مقدار قبل می‌شود.

انواع داده در C#:

نوع داده	کاربرد	میزان حافظه (بایت)	کم‌ترین مقدار	بیش‌ترین مقدار
sbyte	اعداد صحیح	۱	-۱۲۸	۱۲۷
byte	اعداد صحیح مثبت	۱	۰	۲۵۵
short	اعداد صحیح	۲	-۳۲۷۶۸	۳۲۷۶۷
ushort	اعداد صحیح مثبت	۲	-۳۲۷۶۸	۳۲۷۶۷
int	اعداد صحیح	۴	۲- میلیارد	۲ میلیارد
uint	اعداد صحیح مثبت	۴	۰	۴ میلیارد
long	اعداد صحیح	۸	۹- تریلیون	۹ تریلیون
ulong	اعداد صحیح مثبت	۸	۰	۱۸ تریلیون
float	اعداد اعشاری	۴	$-3/4 \times 10^{38}$	$3/4 \times 10^{38}$
doulde	اعداد اعشاری با دقت زیاد	۸	$-1/8 \times 10^{308}$	$1/8 \times 10^{308}$
decimal	اعداد صحیح بزرگ اعداد اعشاری با دقت بسیار بزرگ	۱۶	$-7/9 \times 10^{28}$	$7/9 \times 10^{28}$
bool	مقدار منطقی	۱	false	true
char	کاراکتری	۲	۰	۶۵۵۳۵
string	رشته‌ای	-	-	-
object	آدرس یک داده	-	-	-

داده‌های عددی

داده‌های غیر عددی

روش اعلان متغیر

برای تعریف و ایجاد یک متغیر از شکل کلی مقابل استفاده می‌شود:

نام یک یا چند متغیر نوع داده

```
int    a,b;
byte   x;
```

◆ نکته: نوع متغیر سه ویژگی را مشخص می‌کند که عبارت‌اند از:

- ۱- ظرفیت متغیر
 - ۲- نوع اطلاعاتی که می‌توان در آن ذخیره کرد.
 - ۳- نوع عملیاتی که می‌توان بر روی آن انجام داد.
- بنابراین هر نوع داده، مجموعه‌ای از مقادیر به همراه مجموعه‌ای از عملیات را مشخص می‌کند.

مقداردهی متغیرها:

برای این منظور می‌توان از دستور انتسابی به شکل کلی زیر استفاده کرد:

مقدار = نام متغیر

و یا در زمان تعریف متغیر، آن را مقداردهی اولیه کرد به صورت زیر:

```
مقدار اولیه = نام متغیر نوع داده
int    a = 20;
```

مثال:

◆ نکته: اگر مقدار یا عددی که در یک متغیر قرار داده می‌شود خارج از محدوده‌ی آن باشد، مترجم پیغام خطا می‌دهد. مثلاً دستور `byte a = 256;` خطاست.

◆ نکته: در زبان C# می‌توان اعداد صحیح را در مبنای ۱۶ (هگزا دسیمال) نوشت. بدین منظور قبل از عدد مورد نظر از `0x` یا `0X` استفاده می‌شود.

Byte a = 0XFF;

◆ مثال: دستور مقابل عدد ۲۵۵ را در متغیر a قرار می‌دهد.

◆ کانون فرهنگی آموزش ◆

پسوند نوع داده:

برای مشخص کردن انواع عددی از علائم زیر در انتهای عدد استفاده می‌شود.

نوع داده	نشانه	مثال
uint	u یا U	125U
long	l یا L	1700L
ulong	UL	250000UL
float	f یا F	2.5f
double	d یا D	12.75d
decimal	m یا M	1234567M

◆ نکته: اعداد اعشاری بدون نشانه در حالت پیش فرض از نوع double در نظر گرفته می‌شوند.

با توجه به این که اعداد اعشاری بدون نشانه از نوع double در نظر گرفته می‌شوند، برای تعیین نوع float برای عدد اعشاری باید از پسوند f یا F استفاده شود. بنابراین دستور مقابل خطاست:

```
float a;  
a = 3.5;
```

و شکل صحیح آن به صورت زیر می‌باشد:

```
float a;  
a = 3.5f;
```

قوانین نام‌گذاری متغیرها:

- ۱- نام متغیر نباید با عدد شروع شود.
- ۲- می‌توان از حروف کوچک یا بزرگ، ارقام و علامت خط زیر استفاده کرد.
- ۳- استفاده از کلمات کلیدی یا رزرو شده مجاز نیست.
- ۴- استفاده از فاصله‌ی خالی و سایر علائم مجاز نیست.
- ۵- حروف کوچک و بزرگ یکسان نیست مثلاً متغیر a و متغیر A دو متغیر مستقل هستند.

روش‌های نام‌گذاری متغیرها:

- ۱- روش کوهان شتری: در این روش اگر نام متغیر از چند کلمه تشکیل شده است، حرف اول کلمه‌ی دوم را با حروف بزرگ می‌نویسند. مثل:
passWord , fileName , userName
- ۲- روش مجارستانی: در ابتدای نام متغیر، مخفف نوع داده ذکر می‌شود مثل:

IntNumber , BlnStatus , LngSalary

◆ نکته: در روش نماد علمی هر عدد اعشاری از دو بخش تشکیل می‌شود به نام مانتیس و نما. مانتیس بین ۱ تا ۹ است (فقط یک رقم صحیح دارد) و نما که توانی از ۱۰ است مثل:

$$1 \times 10^3 = 1000$$

$$47.98 = 4.798 \times 10^1$$

$$0.0000265 = 2.65 \times 10^{-5}$$

$$-5600 = -5.6 \times 10^3$$

در زبان C# از فرم نقطه شناور (Floating Point) برای نمایش اعداد اعشاری استفاده می‌شود که مانتیس و نما با حرف E از هم جدا می‌شوند. مثل:

$$0.15 = 1.5E - 1$$

$$-1.482 = -1.482E0$$

$$7800.0 = 7.8E3$$

$$0.0000453 = 4.53E - 5$$

دقت اعداد قابل نمایش:

حداکثر تعداد ارقام غیر صفر مانتیس را دقت عدد می‌نامند. دقت اعداد float با ۶ یا ۷ رقم و دقت اعداد double ۱۵ رقم می‌باشد.

نوع داده کاراکتری:

به هر یک از حروف، ارقام و علائم یک کاراکتر گفته می‌شود. یک کاراکتر داخل دو علامت تک کوتیشن ' ' قرار می‌گیرد. مثل 'A' یا '&' یا '4' و فقط شامل یک کاراکتر می‌باشد.

◆ کانون فرهنگی آموزش ◆

برای مشخص کردن یک کاراکتر می‌توان از کد ۴ رقمی در مبنای ۱۶ استفاده کرد و برای این منظور از علامت \backslash یا x به شکل زیر استفاده می‌شود:
'\ u' کد ۴ رقمی

مثال:

با توجه به این که کد اسکی حرف A در مبنای ۱۰ عدد ۶۵ است و ۶۵ در مبنای ۱۶ برابر ۴۱ می‌باشد می‌توان گفت که 'A' برابر است با '\ u0041'

نوع داده رشته‌ای:

داده‌ی رشته‌ای داخل دو علامت دابل کوتیشن " " نوشته می‌شود و می‌تواند شامل هر کاراکتری با هر تعداد نویسه باشد .

◆ نکته: برای الحاق یا اتصال دو یا چند رشته از علامت + استفاده می‌شود.

مثال:

دستورات مقابل عبارت Visual Studio را روی صفحه‌ی خروجی نمایش می‌دهند.

```
string a = "Studio";
string b = "Visual" + b;
System.Console.WriteLine(b);
```

مثال:

حاصل اجرای دستورات زیر چیست؟

```
int x, y;
x= 2; y=3;
Console.WriteLine("x+y="+x+y);
```

با اجرای قطعه کد فوق عبارت $x+y=23$ چاپ می‌شود. داخل پرانتز دو بار علامت + استفاده شده که هر دو عمل الحاق رشته را انجام می‌دهند.

اگر بخواهیم $x+y$ جمع عددی انجام دهد باید آن را به شکل زیر داخل پرانتز قرار دهیم:

```
Console.WriteLine("x+y="+ (x+y));
```

دریافت رشته:

برای این منظور از متد `ReadLine()` استفاده می‌شود که به کاربر اجازه می‌دهد داده‌ی مورد نظر را از طریق صفحه‌کلید وارد کند. این متد در کلاس Console تعریف شده و در فضای نامی System قرار دارد.

مثال:

```
string name;
name = Console.ReadLine();
```

◆ نکته: داده‌ی وارد شده توسط کاربر به صورت یک رشته در حافظه ذخیره می‌شود.

◆ نکته: متد `ReadLine()` شبیه متد `ReadKey()` است. با این تفاوت که متد `ReadKey()` فقط منتظر دریافت یک کلید می‌باشد اما در `ReadLine()` تا هنگامی که کلید Enter زده نشده منتظر می‌ماند.

تفاوت متد `Read()` با `ReadLine()`:

هر دو برای دریافت یک رشته از کاربر به کار می‌روند. با این تفاوت که متد `Read()` پس از دریافت رشته در همان سطر باقی می‌ماند ولی با `ReadLine()` پس از دریافت رشته مکان‌نما در صفحه‌ی خروجی به سطر بعد منتقل خواهد شد.

تفاوت متد `Write()` با `WriteLine()`:

متد `Write` پس از نمایش اطلاعات در همان سطر مانده و داده یا اطلاعات بعدی را در همان سطر چاپ می‌کند ولی `WriteLine()` پس از چاپ داده‌ها مکان‌نما را به سطر بعد منتقل می‌کند.

مثال:

با اجرای دستورات مقابل عبارت زیر در خروجی نمایش داده می‌شود.

```
Console.Write("ali");
Console.WriteLine("reza");
Console.WriteLine("Hello");
```

alireza
Hello

دریافت اعداد:

با توجه به این که داده‌های وارد شده به وسیله‌ی متد `ReadLine()` به صورت رشته در نظر گرفته و ذخیره می‌شود باید رشته‌ی وارد شده را به عدد تبدیل کنیم. برای این منظور از متد `Parse()` استفاده می‌شود.

◆ نکته: به عمل بررسی کاراکتر به کاراکتر یک رشته برای جدا کردن و به‌دست آوردن یک مقدار با معنی، تجزیه کردن می‌گویند.

مثال:

```
int a = int.Parse("255");
float b = float.Parse("2.5");
```

با اجرای دستور مقابل رشته 255 به عدد صحیح 255 تبدیل و در متغیر a قرار می‌گیرد.
با اجرای دستور مقابل رشته 2.5 به عدد اعشاری 2.5 تبدیل و در متغیر اعشاری b قرار می‌گیرد.

مثال:

کانون فرهنگی آموزش

رشته‌ی input به عدد تبدیل و در متغیر اعشاری number قرار می‌گیرد.

```
string input;
float number;
input = Console.ReadLine();
number = float.Parse(input);
```

البته قطعه کد فوق را به شکل زیر نیز می‌توان نوشت:

```
float number;
input = float.Parse(Console.ReadLine());
```

تست‌های فصل چهارم

- ۳۱- میزان حافظه‌ی کدام نوع داده با بقیه متفاوت است؟
int (۱) unit (۲)
- ۳۲- فضای اشغالی مجموعه متغیرها چند بایت است؟
۳۲ (۱) ۲۴ (۲)
- ۳۳- نام کدام یک از متغیرهای زیر در C# مجاز است؟
2A (۱) x+y (۲)
- ۳۴- ترتیب اجرای عملیات در عبارت روبه‌رو کدام است؟ (از چپ به راست)
A%B - 4 / (X+1) * 2
1 2 3 4 5
۵-۳-۱-۴-۲ (۴) ۱-۳-۵-۴-۲ (۳) ۴-۵-۱-۳-۲ (۲) ۴-۱-۳-۵-۲ (۱)
- ۳۵- حداکثر مقداری که به نوع داده ushort می‌توان نسبت داد، کدام است؟
۳۲۷۶۷ (۴) ۶۵۵۳۵ (۳) ۲۵۵ (۲) ۱۲۷ (۱)
- ۳۶- تعریف و مقداری که کدام متغیر صحیح است؟
sbyte b = -10; (۲) unit a = -5; (۱)
- ۳۷- خروجی دستورات مقابل در چند سطر نمایش داده می‌شود؟
Console.WriteLine(1);
Console.WriteLine(2);
Console.WriteLine ();
Console.WriteLine (3);
Console.WriteLine(4);
۵ سطر (۴) ۳ سطر (۳) ۴ سطر (۲) ۲ سطر (۱)
- ۳۸- اگر بخواهیم رشته‌ی دریافت شده توسط متد ReadLine() را به عدد تبدیل کنیم از کدام متد استفاده می‌شود؟
Convert() (۴) Parse() (۳) float.Parse() (۲) int.Parse() (۱)
- ۳۹- کدام متد فقط منتظر دریافت یا زده شدن یک کلید به وسیله کاربر می‌شود؟
Read() (۳) ReadKey() (۲) ReadLine() (۱)
- ۴۰- خروجی قطعه‌ی کد مقابل چیست؟
int a= 4;
int b= 5;
Console.WriteLine("C="+a+b);
Console.WriteLine("C="+a+b);
C = 9 (۴) C = 45 (۳) C = 49 C = 9 (۲) C = 9 C = 9 (۱)

فصل ۵: عبارت‌های محاسباتی

تعریف عبارت:

به هر یک از عبارت‌های محاسباتی زیر یک عبارت می‌گویند:

$$5.6 * 3 + 5.6$$

$$9 - 7.25$$

$$-8 + 3 * 5$$

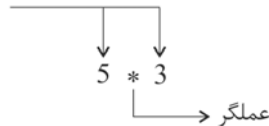
تعریف عملگر:

به هر یک از علائم +، * یا - که بیان‌گر انجام یک عمل روی اعداد و داده‌ها هستند عملگر گفته می‌شود.

تعریف عملوند:

به اعدادی که یک عملگر روی آن‌ها عملی انجام می‌دهد عملوند می‌گویند.

عملوند



♦ نکته: به عملگرهایی که روی دو عدد عمل می‌کنند عملگر دوتایی گفته می‌شود و به عملگرهایی که فقط دارای یک عملوند هستند عملگر یکتایی یا یکانی می‌گویند.

عملگرهای ریاضی یا مسابی:

به ترتیب اولویت عبارت‌اند از:

مثبت و منفی (قرینه‌سازی)	- +	۱
ضرب، تقسیم و باقی‌مانده‌ی تقسیم	%, / *	۲
جمع و تفریق	- +	۳

♦ نکته: پرانتز عملگر نیست ولی اولویت عملگر را بالا می‌برد یعنی ابتدا عبارات داخل پرانتز محاسبه می‌شود.

♦ نکته: در مورد عملگرهای با اولویت یکسان مثل /، %، ابتدا عملگر سمت چپ در عبارت انجام می‌شود.

♦ نکته: در مورد پرانتزهای تو در تو، از داخلی‌ترین پرانتز و در مورد پرانتزهای کنار هم از پرانتز سمت چپ شروع می‌کنیم.

♦ نکته: اگر عملوندهای عملگر / هر دو صحیح باشند حاصل عددی صحیح خواهد بود و اگر یکی از عملوندها اعشاری باشند حاصل عددی اعشاری می‌باشد.

مثال:

$$23 / 5 = 4 \qquad 9 / 2 = 4$$

$$23 / 5.0 = 4.6 \qquad 9.0 / 2 = 9 / 2.0 = 4.5$$

$$23\%5 = 3 \qquad 7.5\%2 = 1.5$$

$$36\%0.2 = 0 \qquad 25419\%10 = 9$$

مثال:

حاصل عبارت زیر را به‌دست آورید.

$$8 + 3 * 5 + 25 \% 4 = 24$$

♦ نکته: اگر حاصل یک عبارت صحیح باشد می‌توان آن را در متغیری از نوع صحیح که گنجایش آن مساوی یا بزرگ‌تر از حاصل عبارت باشد قرار داد.

کانون فرهنگی آموزش

◆ نکته: اگر حاصل یک عبارت صحیح باشد، می‌توان آن را در متغیری از نوع اعشاری قرار داد. به طوری که اعداد بزرگ در نوع float با ۷ رقم و در نوع double با ۱۵ رقم دقت گرد می‌شوند.

◆ نکته: اگر حاصل یک عبارت از نوع اعشاری باشد نمی‌توان آن را در یک متغیر صحیح قرار داد.

◆ نکته: اگر حاصل یک عبارت از نوع double باشد فقط در متغیر اعشاری از نوع double قرار می‌گیرد.

مثال:

float a=19.5 / 4

دستور مقابل خطاست. چون حاصل عبارت 19.5/4 از نوع double می‌باشد و نمی‌توان در متغیر float قرار داد اگر بخواهیم مترجم حاصل عبارت را از نوع float در نظر بگیرد، به شکل زیر عمل می‌کنیم:

float a = 19.5f / 4;

عملگرهای افزایشی و کاهشی:

عملگرهای یکتایی + + و - - برای افزایش و کاهش مقدار متغیر به اندازه‌ی یک واحد در نظر گرفته شده است. این عملگرها را می‌توان قبل یا بعد از نام متغیر به کار برد که بسته به نوع کاربرد می‌تواند مفهوم یکسان یا متفاوتی داشته باشد.

مثال:

در این مثال عملکرد سطر دوم و سوم کاملاً یکسان است و هر کدام یک واحد به X می‌افزایند و در خروجی عدد ۷ چاپ می‌شود.

int x = 5;

x++;

++x;

Console.WriteLine(x);

مثال:

int x = 5, y = 0, z;

y = x++;

z = ++x;

Console.WriteLine(x);

Console.WriteLine(y);

Console.WriteLine(z);

سطر دوم مقدار فعلی X یعنی ۵ را در Y قرار داده، سپس یک واحد به X می‌افزاید. سطر سوم ابتدا یک واحد به X افزوده سپس مقدار جدید را در Z قرار می‌دهد.

بنابراین مقادیر X، Y و Z به ترتیب 7، 5 و 7 خواهد بود.

عملگرهای انتسابی

عبارت‌اند از:

= += -= *= /= %=
&= |= ^= <<= >>=

مثال: عبارت x=5 مقدار 5 را در متغیر X قرار می‌دهد.

مثال: عبارت x+=2 یعنی به متغیر X، ۲ واحد افزوده و حاصل در X قرار گیرد.

مثال: عبارت x*=y یعنی مقدار Y در X ضرب و حاصل در متغیر X قرار گیرد.

مثال: برای سه برابر کردن مقدار متغیر X می‌توان عبارت x*=3 را به کار برد.

تست‌های فصل پنجم

۴۱- کدام دستور یک واحد به مقدار X می‌افزاید؟

(۴) هر سه مورد

(۳) x+=1;

(۲) x=x+1;

(۱) x++;

◆ کانون فرهنگی آموزش ◆

۴۲- حاصل عبارت $55 / ((3\%(4 - (7\%3) + 2) - 5) * 3)$ کدام است؟

- (۱) -3 (۲) -9 (۳) 9 (۴) 3

۴۳- معادل ریاضی عبارت $4 * x + ((5 * x) / (y + 3))$ کدام است؟

- (۱) $4x + \frac{5x}{y+3}$ (۲) $\frac{4x+5x}{y+3}$ (۳) $\frac{4x+5x}{y} + 3$ (۴) $\frac{4x+5x+3}{y}$

۴۴- حاصل عبارت $-12 * 2\%8 - 3 * 2$ کدام است؟

- (۱) -3 (۲) -6 (۳) 6 (۴) 3

۴۵- حاصل تمام گزینه‌های زیر برابر با 1.5 است به جز:

- (۱) 7.5%2 (۲) 15/10 (۳) 150/10.0 (۴) 3/2.0

۴۶- حاصل اجرای دستورات زیر چاپ کدام مورد است؟

int x = 3; int y;

x ++; y = 4;

x += 2;

y += x;

y = x ++;

Console.WriteLine("N = " + x + y);

- (۱) N = 710 (۲) N = 76 (۳) N = 77 (۴) N = 106

$(x \% y * 2) - (43 + (5 / 3))$

۴۷- در عبارت مقابل اولین عملگری که اجرا می‌گردد کدام است؟

- (۱) % (۲) * (۳) / (۴) +

۴۸- کدام یک از انواع داده‌های زیر فقط اعداد صحیح مثبت را قبول می‌کند؟

- (۱) float (۲) ulong (۳) sbyte (۴) int

۴۹- با توجه به دستور زیر کدام گزینه صحیح است؟

int a = 4;

long b = 7;

- (۱) b = 4 / 1.0; (۲) a = b; (۳) b = b / a; (۴) a = a * 1E10;

۵۰- کدام یک از عبارات زیر صحیح است؟

- (۱) byte a = 258; (۲) float b = 29.5 / 4; (۳) double c = 29.5 / 4; (۴) int x = y = z = 5;

فصل ۶: دستورهای شرطی

عبارت منطقی یا بولین

عبارت منطقی عبارتی است که حاصل آن true یا false است.

عملگرهای مقایسه‌ای

این عملگرها می‌توانند دو عدد صحیح یا اعشاری و یا دو داده کاراکتری یا رشته‌ای را با هم مقایسه کنند و نتیجه‌ی آن‌ها یکی از دو مقدار true یا false می‌باشد. عملگرهای مقایسه‌ای عبارت‌اند از:

= = برابری یا مساوی با

= ! نامساوی یا مخالف

< کوچک‌تر از

< بزرگ‌تر از

< = کوچک‌تر یا مساوی با

> = بزرگ‌تر یا مساوی با

عملگرهای منطقی

عملگرهای منطقی به ترتیب اولویت عبارت‌اند از:

اولویت	عملگر	نام عملگر	عملکرد
۱	!	نقیض	عبارت داده شده را معکوس می‌کند.
۲	&&	و	جواب آن زمانی true است که هر دو عملوند true باشد.
۳		یا	جواب آن زمانی false است که هر دو عبارت false باشد.
۴	^	یا انحصاری (XOR)	جواب آن زمانی true است که ارزش دو عبارت مخالف هم باشند.

دستور شرطی if

برای کنترل اجرای دستورها و بررسی شرط، از دستور if استفاده می‌شود و شکل کلی آن به صورت زیر است:

(عبارت منطقی) if

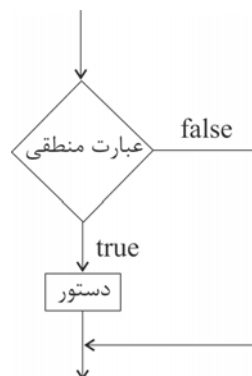
; دستور یا (عبارت منطقی) if

; دستور

◆ نکته: عبارت منطقی باید داخل پرانتز نوشته شود.

◆ نکته: اگر ارزش عبارت منطقی true باشد، دستور مورد نظر اجرا می‌شود و اگر false باشد دستور اجرا نخواهد شد.

فلوچارت دستور if به صورت زیر است:



◆ کانون فرهنگی آموزش ◆

دستور شرطی if – else

کلمه else یعنی در غیر این صورت، شکل کلی این دستور به صورت زیر است:

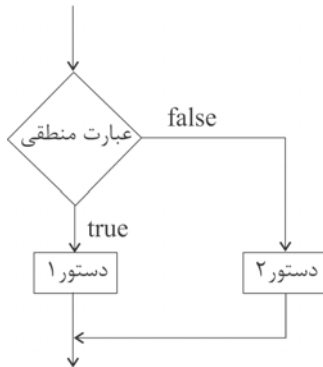
عبارت منطقی) if

دستور ۱ ;

else

دستور ۲ ;

اگر نتیجه‌ی عبارت منطقی true باشد، دستور شماره ۱ و در غیر این صورت، اگر نتیجه‌ی عبارت false باشد، دستور ۲ اجرا می‌شود. فلوجارت دستور if-else به صورت زیر است:



تعریف بلاک

به تعدادی دستور که داخل علامت‌های آکولاد باز و بسته قرار داده باشند بلاک گفته می‌شود. البته بلاک می‌تواند خالی و یا فقط شامل یک دستور باشد.

```
{
دستور ;
دستور ;
دستور ;
}
```

◆ نکته: در دستور شرطی if و یا دستور if-else اگر بخواهیم بیش از یک دستور اجرا شود، از بلاک استفاده می‌نماییم. نتیجه‌ی هر یک از عملگرهای منطقی روی عبارات منطقی به صورت زیر است:

!true=false !false=true	true true=true true false=true false true=true false false=false
true && true = true true && false = false false && true = false false && false = false	true ^ true = false true ^ false = true false ^ true = true false ^ false = false

ساختار if-else پیچیده

در این ساختار اگر شرط اول درست باشد دستور ۱ اجرا می‌شود و اگر شرط اول false باشد بدون اجرای دستور اول شرط دوم بررسی می‌شود و چنانچه شرط دوم درست باشد دستور ۲ اجرا می‌شود و به همین ترتیب الی آخر. اگر هیچ یک از شرطها درست نباشد دستور n ام یا دستور بعد از else آخر اجرا می‌شود. شکل کلی این ساختار به صورت زیر است:

```

if (شرط ۱)
{
    دستورات ۱
}
else if (شرط ۲)
{
    دستورات ۲
}
else if (شرط ۳)
{
    دستورات ۳
}
.
.
else
{
    دستورات n
}

```

دستورات switch

اگر بخواهیم حالت‌های مختلف یک عبارت را بررسی و بر اساس آن دستورهایی را اجرا کنیم از این ساختار استفاده می‌کنیم. اگر مقدار عبارت جلوی **switch** با هر یک از مقدارهای تعیین شده در کلمه‌ی **case** مطابقت داشته باشد، دستور یا دستورات بعد از آن تا رسیدن به کلمه‌ی **break** اجرا می‌شود و اگر با هیچ‌یک از مقادیر **case** برابر نباشد، آن‌گاه دستورات بعد از **default** اجرا می‌شوند.

◆ نکته: بخش **default** و دستورات بعد از آن اختیاری و قابل حذف است.

◆ نکته: نوع داده‌ی عبارت جلوی **switch** نباید از نوع اعشاری باشد. ولی انواع داده‌ی صحیح، حرفی و رشته‌ای قابل قبول است.

شکل کلی دستور **switch** به صورت زیر است:

switch (متغیر یا عبارت)

```

{
    مقدار ۱ :
    دستور ۱
    break ;
    :
    :
    default:
    دستورات n
    break ;
}

```

◆ نکته: برای هر **case**، می‌توان بیش از یک دستور نوشت و نیازی به بلاک ندارد.

◆ نکته: اگر دستور **break** نوشته نشود، مترجم برای جلوگیری از اشتباه برنامه‌نویس، خطا می‌دهد.

◆ نکته: از کلمه‌ی **break** برای خاتمه دادن به یک **case** و خروج از ساختار **switch** استفاده می‌شود.

تست‌های فصل ششم

```

if (n >= 0)
if (n < 10)
x ++;
else
    Console.WriteLine("Ok");

```

۵۱- در چه صورت کلمه‌ی **Ok** چاپ خواهد شد؟

- ۱) اگر n در محدوده‌ی ۰ تا ۱۰ باشد.
- ۲) اگر n در محدوده‌ی ۱ تا ۱۰ باشد.
- ۳) اگر n در محدوده‌ی ۰ تا ۹ باشد.
- ۴) اگر n بزرگ‌تر یا مساوی ۱۰ باشد.

◆ کانون فرهنگی آموزش ◆

۵۲- با توجه به دستورهای شرطی زیر، کلمه‌ی NO در چه صورت چاپ می‌شود؟

```
if (n >= 0)
{
if (n < 10)
Console.WriteLine("NO");
}
else
Console.WriteLine ("OK");
```

- (۱) اگر n منفی باشد.
(۲) اگر n در محدوده‌ی ۰ تا ۱۰ باشد.
(۳) اگر n در محدوده‌ی ۱ تا ۹ باشد.
(۴) اگر n در محدوده‌ی ۰ تا ۹ باشد.

۵۳- حاصل عبارات شرطی زیر کدام گزینه است؟

```
if ((4%(2*4))!=0) Console.WriteLine(4+4);else Console.WriteLine(4*4);
if ((-22/5)=-4.4) Console.WriteLine(2!=2);else console.WriteLine(2==2);
```

- (۱) 8 false (۲) 8 true (۳) 16 false (۴) 16 true

۵۴- به ازای مقادیر $x=2$ ، $y=3$ و $z=4$ حاصل عبارت زیر کدام است؟

$((x * y + 5! = y * z) \&\& (y - x \% z > z + 5))$

- (۱) true (۲) false (۳) 1 (۴) صفر

۵۵- معادل عبارت $((A \wedge B) \parallel B)$ کدام است؟

- (۱) $A \&\&B$ (۲) $A \wedge B$ (۳) $A \parallel B$ (۴) $!(A \parallel B)$

۵۶- اگر A، B و C سه عبارت شرطی باشند، در چه شرایطی OK چاپ می‌شود؟

```
if ((A||!B) &&(A||B) &&C)
```

```
Console.WriteLine("OK");
```

(۳) اگر A و C درست باشند. (۴) اگر A و B و C درست باشند.

(۱) اگر A یا C درست باشد. (۲) اگر A درست باشد.

۵۷- در چه صورت حاصل عملگر منطقی "یا انحصاری" false خواهد شد؟

- (۱) هر دو عبارت true یا هر دو عبارت false باشند.
(۲) یکی از دو عبارت true و عبارت دیگر false باشد.
(۳) عبارت اول false و عبارت دوم true باشد.
(۴) عبارت اول true و عبارت دوم false باشد.

۵۸- به ازای چه مقداری برای متغیر x کلمه‌ی Bad چاپ می‌شود؟

```
switch((x%2==0) &&(x%3==0))
{
case false : Console.WriteLine("Good");break;
case true : Console.WriteLine("Bad");break;
}
```

- (۱) 4 (۲) 9 (۳) 12 (۴) 10

۵۹- با اجرای دستورات مقابل مقدار C کدام است؟

```
int a = 5, b, c = 0;
b = (a + 20) % 10; c = 1;
if (((a + 4) % 2) == b) c = 3; else c = 2;
```

- (۱) 0 (۲) 1 (۳) 2 (۴) 3

۶۰- اگر $a=2$ ، $b=2$ و $c=3$ باشند، خروجی عبارت شرطی زیر کدام است؟

```
if (a == b)
if (b == c)
Console.WriteLine("ali");
else Console.WriteLine("reza");
else Console.WriteLine("omid");
```

- (۱) ali (۲) reza (۳) omid (۴) هیچ کدام

فصل ۷: دستورات تکرار (حلقه‌ها)

ملقه‌ی تکرار for

برای اجرای دستورات مورد نظر به دفعات معین از حلقه for استفاده می‌شود. شکل کلی آن به صورت زیر است:

for (تغییر مقدار متغیر ; عبارت منطقی ; مقدار اولیه = نام متغیر) for

دستور ;

به متغیر حلقه‌ی for، شمارنده‌ی حلقه نیز گفته می‌شود. شمارش می‌تواند به صورت صعودی یا نزولی انجام شود. عبارت منطقی شامل یک عبارت شرطی است که تا زمانی که نتیجه‌ی آن درست است، تکرار دستورات حلقه ادامه می‌یابد و در صورت نادرست یا false بودن آن، برنامه از حلقه خارج می‌شود. برای تغییر مقدار متغیر در حلقه افزایشی از عملگر ++ و در حلقه‌ی کاهش‌ی از عملگر -- استفاده می‌شود.

مثال

دستورات زیر اعداد ۱ تا ۱۰ را روی صفحه‌ی نمایش نشان می‌دهد:

```
for (int i = 1; i <= 10; i++)
    Console.WriteLine(i);
```

مثال

دستورات زیر اعداد ۱۰۰ تا ۱ را روی صفحه‌ی نمایش نشان می‌دهد:

```
for (int i = 100; i >= 1; i--)
    Console.WriteLine(i);
```

◆ نکته: برای خروج از حلقه‌ی for می‌توان از دستور break استفاده کرد.

مثال

دستورات زیر اعداد ۱۰ تا ۶ را به صورت نزولی چاپ می‌کند.

```
int a;
for (a = 10; a >= 1; a--)
{
    if (a == 5) break;
    Console.WriteLine(a);
}
```

◆ نکته: هیچ‌یک از سه بخش داخل پرانتز در دستور for اجباری نیستند.

مثال:

دستورات زیر یک حلقه‌ی تکرار بی‌پایان ایجاد می‌کند که بی‌نهایت بار کلمه Loop را چاپ خواهد کرد:

```
for (; ; )
    Console.WriteLine("Loop");
```

◆ نکته: اگر بعد از پرانتز در حلقه‌ی for علامت ; قرار گیرد متغیر یا شمارنده‌ی حلقه تا زمانی که شرط عبارت منطقی درست باشد عمل شمارش را انجام می‌دهد و دستور بعد از for فقط یک بار اجرا می‌شود چون آن دستور خارج از حلقه‌ی for قرار خواهد گرفت.

مثال:

با اجرای دستورات مقابل کلمه‌ی TEST فقط یک بار چاپ می‌شود:

```
for (int i = 1; i <= 10; i++)
    Console.WriteLine("TEST");
```

◆ کانون فرهنگی آموزش ◆

ملقه‌ی شرطی while

برای اجرای دستورات به دفعات نامعین به کار می‌رود. تکرار اجرای دستورات یا خروج از حلقه بستگی به برقراری شرط حلقه دارد. شکل کلی آن به صورت زیر است:

while (عبارت منطقی)

دستور ;

تا زمانی که عبارت منطقی یا همان شرط حلقه **true** باشد، دستور اجرا می‌شود و در صورت نادرست بودن عبارت منطقی برنامه از حلقه خارج خواهد شد.

◆ نکته: اگر بخواهیم بیش از یک دستور اجرا شود آن‌ها را داخل یک بلاک می‌نویسیم.

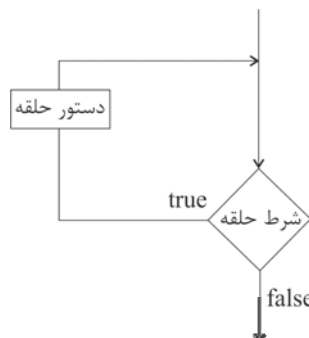
◆ نکته: چنانچه بعد از پرانتز حلقه‌ی **while** علامت ; قرار داده شود اگر شرط حلقه **true** باشد، برنامه در حلقه تکرار دائم قرار می‌گیرد و دستور حلقه هرگز اجرا نمی‌شود و اگر شرط حلقه **false** باشد دستور حلقه فقط یک بار اجرا خواهد شد.

◆ نکته: در حلقه‌ی **while** اگر در همان اولین ورود به حلقه، حاصل عبارت منطقی **false** باشد برنامه از حلقه خارج می‌شود بنابراین ممکن است دستور حلقه هرگز اجرا نشود.

مثال:

با اجرای دستورات زیر اعداد ۱ تا ۱۰ در صفحه نمایش چاپ می‌شوند.

```
int a; a = 1;
while (a <= 10)
{
    Console.WriteLine(a);
    a ++;
}
```



فلوچارت حلقه‌ی شرطی **while** به صورت زیر می‌باشد:

ملقه‌ی شرطی do - while

در این حلقه‌ی شرطی، عبارت منطقی یا شرط حلقه در انتهای حلقه نوشته می‌شود. بنابراین دستور یا دستورات حلقه حداقل یک بار اجرا می‌شود. شکل کلی آن به صورت زیر است:

do

دستور ;

while (عبارت منطقی) ;

کامپایلر با رسیدن به **do** وارد حلقه شده دستور حلقه را اجرا می‌کند و اگر شرط حلقه **true** باشد به دستور **do** برمی‌گردد و اگر **false** باشد کنترل برنامه به سطر بعد از **while** منتقل می‌شود.

◆ نکته: اگر بخواهیم بیش از یک دستور اجرا شود آن‌ها را داخل یک بلاک می‌نویسیم.

مثال:

دستورات زیر اعداد ۱ تا ۱۰ را در صفحه نمایش چاپ می‌کنند.

```
int a = 1;
do
{
    Console.WriteLine(a);
    a ++;
}
while (a <= 10);
```

تست‌های فصل هفتم

۶۱- قطعه کد مقابل چه عملی را انجام می‌دهد؟

```
while (n != 0)
{
x = n % 10;
n = n / 10;
Console.WriteLine(x);
}
```

(۲) چاپ وارون x
(۴) چاپ تعداد ارقام عدد x

(۱) چاپ مقلوب n
(۳) چاپ تعداد ارقام عدد n

۶۲- خروجی برنامه مقابل چاپ کدام عدد است؟

```
int i, j, x = 0;
for (i = 1; j <= 5; i++)
for (j = 1; j <= i; j++)
x = x + 1;
Console.WriteLine(x);
```

(۱) ۲۵
(۲) ۱۵
(۳) ۵
(۴) پیغام خطا

۶۳- در حلقه‌های تو در تو زیر، چند بار علامت * چاپ می‌شود؟

```
for (i = -1; i <= 10; i++)
for (j = 7; j >= i; j--)
Console.WriteLine("*");
```

(۲) ۵۵
(۴) ۷۸

(۱) ۴۵
(۳) ۷۰

۶۴- مقدار نهایی متغیر x کدام است؟

```
int i, j, x = 0;
for (i = 1; i <= 10; i++)
for (j = 1; j <= 10; j++)
x++;
```

(۲) 10
(۴) 20

(۱) 1
(۳) 100

۶۵- خروجی دستورات مقابل چیست؟

```
byte a;
for (a = 5; a >= 1; a++)
Console.WriteLine(a);
```

(۱) اعداد ۵ تا ۱ به صورت نزولی
(۲) اعداد ۱ تا ۵ به صورت صعودی
(۳) اعداد ۵ تا ۲۵۵ به صورت صعودی
(۴) حلقه‌ی for اجرا نمی‌شود.

◆ کانون فرهنگی آموزش ◆

۶۶- خروجی حلقه‌ی for زیر کدام است؟

```
byte i;
for (i = 1; i >= 1; i++)
    Console.WriteLine(i);
```

- (۱) عدد ۱
(۲) اعداد ۱ تا ۲۵۵
(۳) حلقه‌ی تکرار بی‌پایان
(۴) هیچ چیز چاپ نمی‌شود.

۶۷- حاصل اجرای دستورات مقابل چاپ کدام اعداد است؟

```
byte i;
for (i = 1; i >= 1; i--)
    Console.WriteLine(i);
```

- (۱) عدد ۱
(۲) اعداد ۱ تا ۲۵۵ به صورت صعودی
(۳) اعداد ۲۵۵ تا ۱ به صورت نزولی
(۴) حلقه‌ی for به این صورت پیغام خطا صادر می‌کند.

۶۸- خروجی برنامه‌ی مقابل چیست؟

```
byte x = 4;
do
{
    x *= 2;
    Console.WriteLine(x + " ");
}
while (x <= 32);
```

- (۱) 8
(۲) 8 16
(۳) 8 16 32
(۴) 8 16 32 64

۶۹- خروجی دستورات مقابل چیست؟

```
int x = 9;
do
{ Console.WriteLine(x + " ");
  x += 2; }
while (x <= 12);
while (!(x < 12))
{ x -= 2;
  Console.WriteLine(x + " ");
}
```

- (۱) 9 11
(۲) 9 11 9
(۳) 9 11 11
(۴) 9 11 13

۷۰- حاصل اجرای حلقه‌های تو در تو زیر چیست؟

```
int i;
for (i = 1; i <= 10; i++)
    for (i = 1; i <= 10; i++)
        Console.write(i + " ");
```

- (۱) چاپ اعداد ۱ تا ۱۰۰
(۲) چاپ اعداد ۱ تا ۱۰
(۳) دو بار اعداد ۱ تا ۱۰ چاپ می‌شود.
(۴) به دلیل استفاده از متغیر i در هر دو حلقه پیغام خطا صادر می‌شود.

پاسخ تست‌های فصل اول

- ۱- گزینه‌ی «۲»
۲- گزینه‌ی «۴»
۳- گزینه‌ی «۳»
۴- گزینه‌ی «۱»
به نتایج حاصل از انجام محاسبات بر روی داده‌ها، اطلاعات گفته می‌شود.
۵- گزینه‌ی «۳»
دانش می‌تواند مبنای تصمیم‌گیری برای انجام کاری شود.
۶- گزینه‌ی «۲»
زبان C نوعی زبان سطح میانی است.
۷- گزینه‌ی «۴»
۸- گزینه‌ی «۱»
۹- گزینه‌ی «۲»
۱۰- گزینه‌ی «۲»

پاسخ تست‌های فصل دوم

- ۱۱- گزینه‌ی «۲»
۱۲- گزینه‌ی «۳»
۱۳- گزینه‌ی «۴»
۱۴- گزینه‌ی «۱»
۱۵- گزینه‌ی «۳»
۱۶- گزینه‌ی «۲»
۱۷- گزینه‌ی «۴»
۱۸- گزینه‌ی «۲»
۱۹- گزینه‌ی «۳»
۲۰- گزینه‌ی «۲»
برای این منظور باید عبارت using System; در ابتدای برنامه نوشته شود.

پاسخ تست‌های فصل سوم

- ۲۱- گزینه‌ی «۴»
۲۲- گزینه‌ی «۱»
۲۳- گزینه‌ی «۳»
۲۴- گزینه‌ی «۲»
۲۵- گزینه‌ی «۱»
۲۶- گزینه‌ی «۴»

۲۷- گزینه‌ی «۱»

پارامتر اول مدت زمان بر حسب میلی‌ثانیه و پارامتر دوم فرکانس بر حسب هرتز است.

۲۸- گزینه‌ی «۴»

۲۹- گزینه‌ی «۲»

۳۰- گزینه‌ی «۳»

CSC نام مترجم زبان C# است.

پاسخ تست‌های فصل چهارم

۳۱- گزینه‌ی «۴»

داده long ۸ بیتی و سایر گزینه‌ها ۴ بیتی هستند.

۳۲- گزینه‌ی «۲»

متغیر a ۴ بایت، b و c هر کدام ۸ بایت و متغیر d، ۴ بایت نیاز دارد.

۳۳- گزینه‌ی «۳»

۳۴- گزینه‌ی «۱»

۳۵- گزینه‌ی «۳»

۳۶- گزینه‌ی «۲»

نوع اعداد منفی قبول نمی‌کند. در گزینه‌ی ۳، true باید با

حروف کوچک نوشته شود. داده‌ی کاراکتری داخل دو علامت

تک کوتیشن قرار می‌گیرد.

۳۷- گزینه‌ی «۳»

خروجی به صورت زیر خواهد بود:

12
3
4

۳۸- گزینه‌ی «۳»

۳۹- گزینه‌ی «۲»

۴۰- گزینه‌ی «۲»

پاسخ تست‌های فصل پنجم

۴۱- گزینه‌ی «۴»

۵۳- گزینه‌ی «۲»

$$(4\% (2 * 4))! = 0 \xrightarrow{\text{حاصل}} \text{true}$$

$$(-22 / 5) == -4.4 \xrightarrow{\text{حاصل}} \text{false}$$

۵۴- گزینه‌ی «۱»

$$((x * y + 5! = y * z) \& \& (y - x \% z > z + 5))$$

11 true false false

۵۵- گزینه‌ی «۳»

A	B	(A^B)	(A^B) B
T	T	F	T
T	F	T	T
F	T	T	T
F	F	F	F

۵۶- گزینه‌ی «۳»

حاصل عبارت $(A || !B) \& \& (A || B)$ برابر با A خواهد شد.

A	B	A !B	A B	(A !B)&&(A B)
T	T	T	T	T
T	F	T	T	T
F	T	F	T	F
F	F	T	F	F

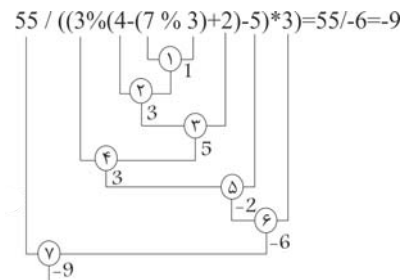
۵۷- گزینه‌ی «۱»

۵۸- گزینه‌ی «۳»

عبارت داخل پرانتز switch یعنی هم بر ۲ و هم بر ۳ بخش‌پذیر باشد که در بین گزینه‌ها فقط عدد ۱۲ هم بر ۲ و هم بر ۳ بخش‌پذیر است.

۵۹- گزینه‌ی «۳»

۴۲- گزینه‌ی «۲»



۴۳- گزینه‌ی «۱»

۴۴- گزینه‌ی «۲»

$$-12 * 2 \% 8 - 3 * 2$$

$$-24 \quad 6$$

$$0 \quad -6$$

۴۵- گزینه‌ی «۲»

حاصل عبارت 15 / 10 برابر با 1 خواهد شد.

۴۶- گزینه‌ی «۲»

X	Y
3	4
4	10
6	6
7	

۴۷- گزینه‌ی «۱»

ابتدا پرانتز سمت چپ محاسبه می‌شود و عملگرهای * و % در یک اولویت قرار دارند به ترتیب از چپ به راست انجام می‌شوند.

۴۸- گزینه‌ی «۲»

۴۹- گزینه‌ی «۳»

۵۰- گزینه‌ی «۳»

پاسخ تست‌های فصل ششم

۵۱- گزینه‌ی «۴»

۵۲- گزینه‌ی «۴»

۶۰- گزینه‌ی «۲»

شرط $a == b$ برابر true است. بنابراین شرط $b == c$ بررسی می‌شود و چون شرط دوم false است دستور بعد از else آن اجرا می‌شود.

پاسخ تست‌های فصل هفتم

۶۱- گزینه‌ی «۱»

کافی است مقدار n را به عنوان مثال ۴۵ در نظر بگیریم.

n	x	چاپ
۴۵	۵	۵۴
۴	۴	
۰		

نکته: عبارت $n \% 10$ نیکان n را برمی‌گرداند.

۶۲- گزینه‌ی «۲»

i	j	x
۱	۱ بار	۱
۲	۲ بار	۳
۳	۳ بار	۶
۴	۴ بار	۱۰
۵	۵ بار	۱۵

۶۳- گزینه‌ی «۱»

$$1+2+3+\dots+9=45$$

i	j
-۱	۹ بار
۰	۸ بار
۱	۷ بار
۲	۶ بار
۳	۵ بار
۴	۴ بار
۵	۳ بار
۶	۲ بار
۷	۱ بار

۶۴- گزینه‌ی «۱»

علامت ; بعد از پرانتز پایان حلقه for را مشخص می‌کند. بنابراین عبارت $x++$ فقط یک بار اجرا می‌شود. اگر هر دو علامت ; از انتهای دستورات for برداشته شود، حاصل مقدار ۱۰۰ خواهد بود.

۶۵- گزینه‌ی «۳»

چون میزان تغییرات شمارنده حلقه، افزایشی است؛ بنابراین عبارت منطقی که پایان حلقه یا خروج از آن را مشخص می‌سازد همواره true خواهد بود و تا آخرین مقدار در محدوده byte عمل شمارش انجام خواهد شد.

۶۶- گزینه‌ی «۲»

بخش دوم حلقه‌ی for یعنی عبارت منطقی $i >= 1$ همواره true خواهد بود.

۶۷- گزینه‌ی «۱»

۶۸- گزینه‌ی «۱»

نقیض عبارت $8 <= 32$ برابر با false خواهد شد و برنامه به دستور do بر نمی‌گردد.

۶۹- گزینه‌ی «۳»

۷۰- گزینه‌ی «۲»