

۱۵۴۵۵۰۰
۹۲۰۸/۱۶

به نام خدا

چگونه با C و C++

برنامه بنویسیم

دیتل - دیتل

ویرایش بین‌المللی

(ویراست ۸)

بهرام پاشایی

۳

سرشناسه	Deitel, Paul J. دیتل، پل ج.، ۱۹۶۹ - م.
عنوان و نام پدیدآور	چگونه با C و C++ برنامه بنویسیم / دیتل - دیتل؛ بهرام پاشایی.
مشخصات نشر	تهران: ادبستان، ۱۳۹۶.
مشخصات ظاهری	۸۴۸ ص.: مصور (رنگی).
شابک	978-600-196-105-2
وضعیت فهرست نویسی	فیا
یادداشت	عنوان اصلی: C : how to program, 8th ed, ©2016
موضوع	سی (زبان برنامه نویسی کامپیوتر) (Computer program language) C
موضوع	سی ++ (زبان برنامه نویسی کامپیوتر) (Computer program language) C++
موضوع	جاوا (زبان برنامه نویسی کامپیوتر) (Computer program language) Java
شناسه افزوده	دیتل، هاروی ام.، ۱۹۴۵ - م. Deitel, Harvey M.
شناسه افزوده	پاشایی، بهرام، ۱۳۵۳ - مترجم
رده بندی کنگره	۷۳/۷ ۱۳۹۶ ۸۴۵۹۳/س QA
رده بندی دیسی	۰۰۵/۱۳۳
شماره کتابشناس ملی	۴۹۴۰۸۹۵

نام کتاب	چگونه با C و C++ برنامه بنویسیم (ویراست ۸)
مؤلفین	Paul Deitel - Harvey Deitel
مترجم	بهرام پاشایی
ناشر	ادبستان
لیتوگرافی	طیف نگار
چاپ	دیا - طنین
نوبت چاپ	اول - ۱۳۹۶
تیراژ	۳۰۰ نسخه
قیمت	۷۰۰۰۰ تومان
شابک	۹۷۸-۶۰۰-۱۹۶-۱۰۵-۲
ISBN	978-600-196-105-2

دفتر مرکزی کتاب آیلار
 انقلاب - خیابان شهید منیری جاوید (اردیبهشت) - خیابان شهدای ژاندارمری شرقی - شماره ۱۴۶
 (ساختمان آیلار)
 فروشگاه شماره ۱ (کتاب آیلار)
 انقلاب - روبروی دبیرخانه دانشگاه تهران - بازارچه کتاب
 فروشگاه شماره ۲ (کتاب آیلار)
 کریمخان زند - مابین ایرانشهر و خردمند جنوبی - شماره ۱۳۲
 تلفن: ۸۸۳۱۹۷۴۰-۱

این اثر مشمول قانون حمایت از حقوق مؤلفان، مصنفان و هنرمندان مصوب ۱۳۴۸ است. هر کس تمام یا قسمتی از این اثر را بدون اجازه ناشر، نشر یا پخش یا الکترونیک عرضه کند تحت پیگرد قانونی قرار خواهد گرفت.

به کتاب چگونه با C و C++ برنامه‌نویسی کنیم خوش آمده‌اید! این کتاب با توجه به جدیدترین تکنولوژی‌های آموزشی برای دانشجویان، اساتید و کسانی تهیه شده است که به صورت حرفه‌ای در توسعه نرم‌افزار نقش دارند.

ویژگی کتاب‌های برادران دیتل، شیوه "live code" است. در این شیوه مفاهیم از طریق برنامه‌های کامل به جای استفاده از تکه کد آموزش داده می‌شوند. به دنبال هر کد مثال یک یا چند نمونه از اجرای برنامه قرار داده شده تا مخاطب شاهد نتیجه اجرای واقعی کد باشد.

برخی از ویژگی‌های این کتاب عبارتند از:

- آمیخته شدن با قابلیت‌های استانداردهای C11 و C99. از طریق کامپایلرهای مختلف از C11 و C99 پشتیبانی شده است. Microsoft Visual C++ از زیر مجموعه متنوعی پشتیبانی می‌کند که به C، C99 و C11 افزوده شده‌اند؛ ویژگی‌ای که این C++ استاندارد لازم هستند. در بخش‌های ابتدایی کتاب به طور گسترده از ویژگی‌های مورد نیاز C11 و C99 استفاده کرده‌ایم، ویژگی‌های که برای تدریس لازم هستند.
- تمام کدها بر روی سیستم‌های Linux, Windows و OS X تست شده‌اند. تمام مثال‌ها و کد تمرینات با استفاده از GNU gcc بر روی Linux، روال C++ بر روی ویندوز (در Visual Studio 2013 Community Edition) و LLVM در Xcode بر روی OS X تست شده‌اند.
- C++ و برنامه‌نویسی شی‌گرا به روز شد. فصل‌های ۱۵ الی ۲۳ متمرکز بر برنامه‌نویسی شی‌گرا در C++ هستند که با توجه به کتاب C++ to program 9/e تهیه و با استاندارد C++11 کاملاً به روز شده‌اند.
- سبک جدید کدنویسی و اعلان متغیرها.
- افزودن توضیحات بیشتر به متن کد.
- استفاده از اصطلاحات فنی استاندارد.

ویژگی‌های دیگر

برخی از ویژگی‌های به کار رفته در بخش برنامه‌نویسی C عبارتند:

- برنامه‌نویسی امن در C: در انتهای بسیاری از فصل‌های C بخشی با عنوان برنامه وسیع امن در C قرار داده شده است.
- تاکید بر بحث کارایی. زبان C و C++ محبوب کسانی هستند که در زمینه طراحی سیستم‌های با کارایی بسیار بالا مانند سیستم‌های عامل، سیستم‌های زمان واقعی، سیستم‌های تعبیه شده و سیستم‌های ارتباطی کار می‌کنند. از این رو سعی کرده‌ایم کارایی در سطح بالا را حفظ کنیم.
- مرتب‌سازی. در فصل ۶ شروع به توضیح مفهوم مرتب‌سازی با یک الگوریتم ساده کرده‌ایم. در ادامه به بررسی چندین الگوریتم و مقایسه آنها از منظر استفاده از حافظه و قدرت پردازنده پرداخته‌ایم. به همین منظور از نماد آشنای Big O استفاده کرده‌ایم. در تمرینات چندین الگوریتم مرتب‌سازی معرفی شده است.



نرم افزار به کار رفته در کتاب

برنامه های کتاب با استفاده از کامپایلرهای زیر که مجانی هم هستند تست شده است:

- GNU C و C++ (<http://gcc.gnu.org/install/binaries.html>) که در حال حاضر بر روی اکثر سیستم های Linux نصب است و می توان آن را بر روی سیستم های OSX و Windows نصب کرد.
 - Visual C++ در Visual Studio 2013 Community edition که می توانید از <http://go.microsoft.com/?linkid=9863608> دانلود کنید.
 - LLVM در Apple's Xcode IDE، که کاربران OS X می توانند از Mac App Store دانلود کنند.
- برای داشتن کامپایلرهای مجانی C و C++ می توانید به آدرس های زیر هم مراجعه کنید:

<http://www.thefreecountry.com/compilers/cpp.shtml>
<http://www.compilers.net/Dir/Compilers/CCpp.htm>
<http://www.freebyte.com/programming/cpp/#cppcompilers>
http://en.wikipedia.org/wiki/List_of_compilers#C.2B.2B_compilers

کامپایلر مورد نظر خود را تهیه و نصب کنید.

تکته: CD همراه کتاب حاوی کامپایلرهای مورد نیاز است.

ایجاد پروژه در Visual Studio

بر روی یک سیستم ویندوز تمام آن چیزی که نیاز دارید نصب Visual Studio Community است. کامپایلر C++ قادر به کامپایل تقریباً تمام کدهای بخش برنامه نویسی با C است.

برای کامپایل یک برنامه C:

۱. ویژوال استودیو را باز کنید.
 ۲. **File > New > Project...** را انتخاب کنید.
 ۳. از سمت چپ کادر تبدالی **New Project**، بخش **Templates** را باز کرده و **Visual C++** را انتخاب کنید.
 ۴. از میانه کادر تبدالی، **Win32 Consol Application** را انتخاب کنید.
 ۵. از پایین کادر تبدالی، نام و مکان ایجاد پروژه را مشخص و بر روی **OK** کلیک کنید.
- در کادر **Win32 Application Wizard**:

۱. بر روی **Next >** کلیک کنید.
۲. هر دو جعبه **Precompiled header** و **Security Development Lifecycle (SDL)** را از انتخاب خارج کنید.
۳. **Empty Project** را انتخاب کنید.
۴. بر روی **Finsih** کلیک کنید.



در ادامه، یک فایل source code به پوشه **Source Files** در **Solution Explorer** که در سمت راست Visual Studio قرار گرفته اضافه کنید. اگر این پنجره دیده نمی‌شود، **View > Solution Explorer** را انتخاب کنید.

- اگر می‌خواهید تا یکی از برنامه‌های این کتاب را اجرا کنید، کافی است به آسانی فایل‌های آن را از پنجره **Windows Explorer (Windows 7)** یا **File Explorer (Windows 8, 10)** به درون پوشه **Source Files** در **Solution Explorer** درآگ کنید. به محض انجام این کار، یا فشردن **Ctrl+F5** برنامه کامپایل و اجرا خواهد شد.

- اگر می‌خواهید یک برنامه را از ابتدا ایجاد کنید (این حالت را برای آموختن برنامه‌های این کتاب توصیه می‌کنیم) بر روی پوشه **Source Files** در **Solution Explorer** کلیک راست انجام داده و سپس **Add > New Item...** را انتخاب کنید. در کادر تبادلی، **C++ file** را انتخاب کنید، اما نام فایل با پسوند **“.c”** و نه **“.cpp”** باشد. با این انتخاب Visual Studio متوجه خواهد شد که برنامه را به عنوان یک برنامه C کامپایل خواهد کرد و نه C++.

کامپایل و اجرا با استفاده از **GNU C++** بر روی **Linux**

خط اعلان در شل (shell) یا پوسته بر روی سیستم از کاراکتر مد (~) استفاده می‌کند تا نشانی از شاخه home باشد، و هر اعلان با یک کاراکتر دلار (\$) خاتمه پیدا می‌کند. خط اعلان در میان سیستم‌های لینوکس بسیار متنوع است. فرض ما در اینجا بر این است که شما در حال حاضر بر یک فایل **c** ایجاد یا در اختیار دارید (مانند مثال‌های این کتاب) که آماده کامپایل شدن است.

گام ۱: یافتن برنامه

در پوسته لینوکس، از دستور **cd** برای تغییر به شاخه حاوی فایل **Number.c** می‌خواهید آن را کامپایل کنید، استفاده نمائید. برای مشخص بودن بحث، ما این فایل فرضی را **Number.c** می‌نامیم و فرض می‌کنیم پوشه **examples** در درون پوشه حساب کاربری شما قرار داشته باشد. همچنین فرض می‌کنیم که **GuessNumber.c** در درون زیرپوشه **ch1** از **examples** قرار دارد.

در پنجره پوسته، تایپ کنید

```
cd examples/ch01
```

سپس **Enter** را فشار دهید. شاهد محتویات درون شاخه با تایپ **ls** و فشردن **Enter** خواهید بود.

گام ۲: کامپایل برنامه

قبل از اجرای برنامه، ابتدا باید آن را با تایپ دستور زیر کامپایل کرده باشید

```
gcc -std=c11 GuessNumber.c -o GuessNumber
```



این دستور، برنامه را بر اساس C11 کامپایل کرده و یک فایل اجرایی به نام **GuessNumber** تولید می‌کند. برای سایر برنامه‌ها کافی است تا **GuessNumber** را با نام برنامه مقتضی جایگزین کنید. اگر برنامه حاوی چندین فایل **c**. باشد و فقط این برنامه در شاخه حضور داشته باشد، می‌توانید از **c*** استفاده کنید تا نشان دهد که تمام فایل‌های **c**. در آن شاخه باید کامپایل شوند.

گام ۳: اجرای برنامه

برای اجرا کردن فایل اجرایی **GuessNumber**، تایپ کنید

`./GuessNumber`

در اعلان به نمایش در آمده، **Enter** را فشار دهید. / به لینوکس می‌گوید تا برنامه را از شاخه جاری اجرا کند و برای نشان دادن این که **GuessNumber** یک فایل اجرایی است لازم می‌باشد.

ایجاد پروژه در Xcode بر روی MacOS

گام ۱: اجرای Xcode

پنجره **Finder** را باز کرده، **Applications** را انتخاب و بر روی آیکون **Xcode** دابل کلیک کنید. اگر **Xcode** برای بار نخست اجرا شود، پنجره **Welcome to Xcode** ظاهر خواهد شد. این پنجره را ببندید. با انتخاب **Window > Welcome to Xcode** می‌توانید به این پنجره دست پیدا کنید.

گام ۲: انتخاب پروژه

۱. **Select File > New > Project...** را انتخاب کنید.
 ۲. در **OS X** (یا **MacOS**) زیر رده **Command Line Tool, Application** انتخاب کرده و بر روی **Next** کلیک کنید.
 ۳. نامی برای پروژه در فیلد **Product Name** وارد کنید.
 ۴. مطمئن شوید که زبان انتخاب شده **C** باشد و سپس بر روی **Next** کلیک کنید.
 ۵. محل ذخیره شدن پروژه را تعیین، و بر روی **Create** کلیک کنید.
- در حالت پیش فرض، **Xcode** یک فایل کد منبع **main.cpp** ایجاد می‌کند که حاوی یک برنامه ساده با هدف نمایش عبارت **"Hello, World!"** است. پنجره به چهار ناحیه اصلی در زیر میله ابزار تقسیم شده است: ناحیه **Navigator** (چپ)، ناحیه **Editor** (مرکز) و ناحیه **Utilities** (راست) که در ابتدای کار ظاهر می‌شوند. ناحیه **Debug** (پایین) هم وجود دارد که برنامه را با آن اجرا می‌کنید.



گام ۳: فایل main.c

می‌توانید از main.c به عنوان فایلی که برنامه جدید خود را در آن تایپ می‌کنید استفاده کنید، یا این فایل را حذف کرده و یک فایل c. موجود را به پروژه اضافه کنید. برای تایپ یک برنامه جدید، بر روی main.c دابل کلیک کنید تا باز شود.

برای افزودن فایل متفاوت دیگری، main.c را انتخاب و حذف کنید، سپس بر روی پوشه‌ای که حاوی main.c بود کلیک راست انجام داده و **Add Files to "folder name"...** را انتخاب کنید. در ادامه می‌توانید به سراغ مکان‌هایی بروید که حاوی فایل یا فایل‌های موجود ذخیره شده هستند و آنها را انتخاب و به پروژه اضافه کنید. همچنین می‌توانید به آسانی فایل‌ها را از پنجره Finder به درون پوشه درآگ کنید.

گام ۴: کامپایل و اجرای پروژه

برای کامپایل و اجرای پروژه کافی است بر روی دکمه run کلیک کنید (که شبیه دکمه play است). این دکمه در سمت چپ میله ابزار Xcode قرار گرفته است. اگر برنامه به درستی کامپایل شود، Xcode ناحیه Debug را باز می‌کند (در پایین ناحیه Editor) و برنامه در نیمه سمت راست ناحیه Debug اجرا می‌شود. اگر برنامه حالت تعاملی داشته باشد، می‌توانید با کلیک بر روی این پنجره، داده‌های مورد نیاز برنامه را وارد کنید.

فهرست مطالب

۲۵	فصل اول : معرفی کامپیوتر، اینترنت و وب
۲۶	۱-۱ مقدمه
۲۶	۱-۲ سخت افزار و نرم افزار
۲۷	۱-۲-۱ قانون مور
۲۷	۱-۲-۲ سازمان های کامپیوتر
۲۸	۱-۳ سلسله مراتب داده
۳۱	۱-۴ زبان ماشین، زبان اسمبلی و زبان های سطح بالا
۳۳	۱-۵ زبان برنامه نویسی
۳۴	۱-۶ کتابخانه استاندارد C
۳۵	۱-۷ C++ و سایر زبان های مبتنی بر
۳۷	۱-۸ تکنولوژی شی
۳۷	۱-۸-۱ اتومبیل به عنوان یک شی
۳۷	۱-۸-۲ متدها و کلاس ها
۳۸	۱-۸-۳ نمونه سازی
۳۸	۱-۸-۴ استفاده مجدد
۳۸	۱-۸-۵ پیغام ها و فراخوانی متد
۳۸	۱-۸-۶ صفات و متغیرهای نمونه
۳۹	۱-۸-۷ کپسوله سازی و پنهان کردن اطلاعات
۳۹	۱-۸-۸ توارث
۳۹	۱-۹ محیط توسعه برنامه C
۳۹	۱-۹-۱ فاز ۱: ایجاد برنامه
۴۰	۱-۹-۲ فازهای ۲ و ۳: پیش پردازش و کامپایل برنامه C
۴۰	۱-۹-۳ فاز ۴: لینک
۴۲	۱-۹-۴ فاز ۵: بار شدن
۴۲	۱-۹-۵ فاز ۶: اجرا
۴۲	۱-۹-۶ مشکلاتی که ممکن است در زمان اجرا رخ دهند
۴۲	۱-۹-۷ ورودی، خروجی استاندارد و استریم استاندارد خط
۴۲	۱-۱۰ تست یک برنامه C در ویندوز، لینوکس و مکینتاش
۴۳	۱-۱۰-۱ اجرای برنامه C از طریق خط دستور ویندوز
۴۹	۱-۱۱ سیستم های عامل
۵۰	۱-۱۱-۱ ویندوز یک سیستم عامل اختصاصی
۵۰	۱-۱۱-۲ لینوکس سیستم عامل open-source
۵۱	۱-۱۱-۳ آندروید



۵۲	۱-۱۲ اینترنت و World Wide Web
۵۳	۱-۱۲-۱ وب گسترده جهانی: اینترنت را کارپسند کرد
۵۳	۱-۱۲-۲ سرویس‌های وب
۵۵	۱-۱۲-۳ Ajax
۵۵	۱-۱۲-۴ IoT
۵۶	۱-۱۳ تکنولوژی‌های نرم‌افزاری
۵۸	۱-۱۴ به روز ماندن با IT
۵۹	خودآزمایی
۶۰	پاسخ خودآزمایی
۶۰	تمرینات
۶۳	فصل دوم: مفاهیم پایه و برنامه‌نویسی C
۶۴	۲-۱ مقدمه
۶۴	۲-۲ یک برنامه ساده د: C چاپ عبارت متنی
۶۸	۲-۳ یک برنامه ساده دی: C جمع دو عدد صحیح
۷۳	۲-۴ مفاهیم حافظه
۷۴	۲-۵ محاسبات در C
۷۷	۲-۶ تصمیم‌سازی: عملگرهای برابر و رابطه‌ای
۸۱	۲-۷ برنامه‌نویسی امن C
۸۳	خودآزمایی
۸۵	پاسخ خودآزمایی
۸۶	تمرینات
۹۱	فصل سوم: توسعه برنامه ساخت یافته در C
۹۲	۳-۱ مقدمه
۹۲	۳-۲ الگوریتم
۹۲	۳-۳ شبه‌کد
۹۳	۳-۴ ساختارهای کنترل
۹۵	۳-۵ عبارت انتخاب if
۹۷	۳-۶ عبارت انتخاب if..else
۱۰۱	۳-۷ عبارت تکرار while
۱۰۲	۳-۸ فرموله کردن الگوریتم: شمارنده-تکرار کنترل شده
۱۰۴	۳-۹ فرموله کردن الگوریتم‌ها: تکرار-کنترل شده مراقبتی
۱۰۹	۳-۱۰ فرموله کردن الگوریتم‌ها: عبارات کنترلی تودرتو
۱۱۳	۳-۱۱ عملگرهای تخصیص‌دهنده
۱۱۴	۳-۱۲ عملگرهای افزایش و کاهش
۱۱۶	۳-۱۳ برنامه‌نویسی امن در C
۱۱۸	خودآزمایی



۱۲۰.....	پاسخ خودآزمایی.....
۱۲۲.....	تمرینات.....
۱۲۹.....	فصل چهارم: کنترل برنامه در C
۱۳۰.....	۴-۱ مقدمه.....
۱۳۰.....	۴-۲ اصول تکرار.....
۱۳۱.....	۴-۳ شمارنده-تکرار کنترل شده.....
۱۳۲.....	۴-۴ عبارت تکرار for.....
۱۳۵.....	۴-۵ عبارت for: نکته‌ها و دیدگاه‌ها.....
۱۳۶.....	۴-۶ مثال‌های با استفاده از عبارت for.....
۱۳۹.....	۴-۷ عبارت switch انتخابی.....
۱۴۵.....	۴-۸ عبارت تکرار do...while.....
۱۴۶.....	۴-۹ عبارات continue و break.....
۱۴۸.....	۴-۱۰ عملگرهای منطقی.....
۱۵۱.....	۴-۱۱ اشتباه گرفتن عملگر برابر (=) و عملگر تخصیص (=).....
۱۵۳.....	۴-۱۲ چیکده برنامه‌نویسی ساخت یافته.....
۱۵۸.....	۴-۱۳ برنامه‌نویسی امن در C.....
۱۶۰.....	خودآزمایی.....
۱۶۱.....	پاسخ خودآزمایی.....
۱۶۳.....	تمرینات.....
۱۶۹.....	فصل پنجم: توابع در C
۱۷۰.....	۵-۱ مقدمه.....
۱۷۰.....	۵-۲ پیمانه‌ای کردن برنامه‌ها در C.....
۱۷۱.....	۵-۳ توابع کتابخانه math.....
۱۷۳.....	۵-۴ توابع.....
۱۷۳.....	۵-۵ تعریف تابع.....
۱۷۳.....	۵-۵-۱ تابع square.....
۱۷۶.....	۵-۵-۲ تابع maximum.....
۱۷۷.....	۵-۶ پیش نمونه‌های تابع: نگاهی عمیق‌تر.....
۱۷۹.....	۵-۷ عملکرد پشته فراخوانی و رکورد ثبت فعالیت.....
۱۸۳.....	۵-۸ سِرَآیندها.....
۱۸۴.....	۵-۹ ارسال آرگومان: با مقدار و با مراجعه.....
۱۸۵.....	۵-۱۰ تولید اعداد تصادفی.....
۱۹۳.....	۵-۱۲ کلاس‌های ذخیره‌سازی.....
۱۹۶.....	۵-۱۳ قوانین قلمرو.....
۱۹۹.....	۵-۱۴ بازگشتی.....
۲۰۳.....	۵-۱۵ مثال بازگشتی: سری فیبوناچی.....



۲۰۵.....	۵-۱۶ بازگشتی در برابر تکرار
۲۰۶.....	۵-۱۷ برنامه‌نویسی امن در C
۲۰۷.....	خودآزمایی
۲۱۰.....	پاسخ خودآزمایی
۲۱۲.....	تمرینات
۲۱۹.....	فصل ششم: آرایه‌ها در C
۲۲۰.....	۶-۱ مقدمه
۲۲۰.....	۶-۲ آرایه‌ها
۲۲۲.....	۶-۳ تعریف آرایه
۲۲۲.....	۶-۴ مثال‌هایی از کاربرد آرایه
۲۲۲.....	۶-۴-۱ آرایه و استفاده از حلقه برای مقداردهی اولیه عناصر آرایه
۲۲۳.....	۶-۴-۲ مقداردهی اولیه آرایه در تعریف با لیست مقداردهی
۲۲۴.....	۶-۴-۳ تعیین سایز آرایه با ثابت سمبولیک و مقداردهی عناصر آرایه با محاسبه
۲۲۵.....	۶-۴-۴ به دست آوردن مجموعه عناصر آرایه
۲۲۶.....	۶-۴-۵ استفاده از آرایه‌ها برای تحلیل نتایج
۲۲۸.....	۶-۴-۶ نمایش گرافیکی داده‌های آرایه توسط نمودارهای میله‌ای
۲۲۹.....	۶-۴-۷ پرتاب شصت میلیون بار طاق و خلاصه نتایج در آرایه
۲۲۹.....	۶-۵ استفاده از آرایه‌های کاراکتری برای ذخیره و دستکاری رشته‌ها
۲۳۰.....	۶-۵-۱ مقداردهی آرایه کاراکتری با رشته
۲۳۰.....	۶-۵-۲ مقداردهی آرایه کاراکتری با لیست رشته‌ها از کاراکترها
۲۳۰.....	۶-۵-۳ دسترسی به کاراکترها در یک رشته
۲۳۰.....	۶-۵-۴ وارد کردن رشته به آرایه کاراکتری
۲۳۱.....	۶-۵-۵ چاپ آرایه کاراکتری حاوی رشته
۲۳۱.....	۶-۵-۶ توصیف عملکرد آرایه‌های کاراکتری
۲۳۲.....	۶-۶ آرایه‌های محلی استاتیک و اتوماتیک
۲۳۴.....	۶-۷ ارسال آرایه به توابع
۲۳۸.....	۶-۸ مرتب‌سازی آرایه‌ها
۲۴۰.....	۶-۹ میث آموزش: محاسبه میانگین، میانه و مد با استفاده از آرایه‌ها
۲۴۴.....	۶-۱۰ جستجوی آرایه‌ها
۲۴۴.....	۶-۱۰-۱ جستجوی خطی
۲۴۵.....	۶-۱۰-۲ جستجوی باینری
۲۴۹.....	۶-۱۱ آرایه‌های چند بعدی
۲۴۹.....	۶-۱۱-۱ نمایش گرافیکی آرایه دو بعدی
۲۵۰.....	۶-۱۱-۲ مقداردهی آرایه دو بعدی
۲۵۲.....	۶-۱۱-۳ تنظیم عناصر در یک سطر
۲۵۲.....	۶-۱۱-۴ مجموع عناصر در آرایه دو بعدی



۲۵۲	۵-۱۱-۶ دستکاری آرایه دو بعدی
۲۵۵	۱۲-۶ آرایه به طول متغیر
۲۵۸	۱۳-۶ برنامه‌نویسی امن در C
۲۶۱	خودآزمایی
۲۶۳	پاسخ خودآزمایی
۲۶۵	تمرینات
۲۷۵	فصل هفتم: اشاره‌گرها در C
۲۷۶	۱-۷ مقدمه
۲۷۶	۲-۷ اعلان و مقداردهی متغیرهای اشاره‌گر
۲۷۷	۳-۷ عملگرهای اشاره‌گر
۲۸۰	۴-۷ ارسال آرمان به توابع به روش مراجعه
۲۸۴	۵-۷ کاربرد const همپای اشاره‌گرها
۲۸۵	۱-۵-۷ تبدیل یک رشته به روف بزرگ با استفاده از اشاره‌گر غیرثابت به داده غیرثابت
۲۸۶	۲-۵-۷ چاپ یک کاراکتر از رشته هر زمان با استفاده از اشاره‌گر غیرثابت به داده ثابت
۲۸۸	۳-۵-۷ اقدام به تغییر یک اشاره‌گر ثابت به داده غیرثابت
۲۸۹	۴-۵-۷ اقدام به تغییر یک اشاره‌گر ثابت به داده ثابت
۲۹۰	۶-۷ مرتب‌سازی حبابی با ارسال به روش مراجعه
۲۹۳	۷-۷ عملگر sizeof
۲۹۵	۸-۷ اشاره‌گر عبارات و محاسبات اشاره‌گر
۲۹۵	۱-۸-۷ عملگرهای مجاز برای محاسبات اشاره‌گر
۲۹۵	۲-۸-۷ اشاره دادن اشاره‌گر به آرایه
۲۹۶	۳-۸-۷ افزودن یک مقدار صحیح به اشاره‌گر
۲۹۶	۴-۸-۷ تفریق یک مقدار صحیح از اشاره‌گر
۲۹۷	۵-۸-۷ افزایش و کاهش اشاره‌گر
۲۹۷	۶-۸-۷ تفریق یک اشاره‌گر از دیگری
۲۹۷	۷-۸-۷ تخصیص اشاره‌گر به دیگری
۲۹۷	۸-۸-۷ اشاره‌گر به void
۲۹۸	۹-۸-۷ مقایسه اشاره‌گرها
۲۹۸	۹-۷ رابطه بین اشاره‌گرها و آرایه‌ها
۲۹۸	۱-۹-۷ نشان‌گذاری اشاره‌گر/افست
۲۹۹	۲-۹-۷ نشان‌گذاری اشاره‌گر/شاخص
۲۹۹	۳-۹-۷ عدم تغییر نام آرایه با محاسبات اشاره‌گر
۲۹۹	۴-۹-۷ توصیف شاخص‌گذاری و افست اشاره‌گر
۳۰۰	۵-۹-۷ کپی رشته‌ها با آرایه‌ها و اشاره‌گرها
۳۰۲	۱۰-۷ آرایه‌ای از اشاره‌گرها
۳۰۳	۱۱-۷ مبحث آموزشی: بازی کارت



۳۰۷	۱۲-۷ اشاره گرهای تابع
۳۰۷	۱۲-۷ مرتب سازی صعودی یا نزولی
۳۱۰	۱۲-۷ استفاده از اشاره گرهای تابع در ایجاد یک سیستم متکی بر منو
۳۱۱	۱۳-۷ برنامه نویسی امن در C
۳۱۳	خود آزمایی
۳۱۵	پاسخ خود آزمایی
۳۱۶	تمرینات
۳۲۱	بخش ویژه: ساخت کامپیوتر متعلق بخود
۳۲۷	فصل هشتم: کاراکترها و رشته ها در C
۳۲۸	۸-۱ مقدمه
۳۲۸	۸-۲ مفاهیم بنیادین رشته ها و کاراکترها
۳۳۰	۸-۳ Character-Handling کتابخانه
۳۳۰	۸-۳-۱ isdigit و isdigit, isalpha, isalnum تابع
۳۳۲	۸-۳-۲ toupper و islower, isupper, tolower تابع
۳۳۳	۸-۳-۳ isgraph و isspace, isctrl, ispunct, isprint تابع
۳۳۴	۸-۴ توابع تبدیل کننده
۳۳۵	۸-۴-۱ strtod تابع
۳۳۵	۸-۴-۲ strtol تابع
۳۳۶	۸-۴-۳ strtoul تابع
۳۳۷	۸-۵ توابع کتابخانه Standard Input/Output
۳۳۷	۸-۵-۱ putchar و fgets تابع
۳۳۹	۸-۵-۲ getchar تابع
۳۳۹	۸-۵-۳ sprintf تابع
۳۴۰	۸-۵-۴ sscanf تابع
۳۴۰	۸-۶ توابع دستکاری کننده رشته از کتابخانه String-Handling
۳۴۱	۸-۶-۱ strcpy و strncpy تابع
۳۴۲	۸-۶-۲ strcat و strncat تابع
۳۴۳	۸-۷ توابع مقایسه کننده از کتابخانه String-Handling
۳۴۴	۸-۸ توابع جستجو کننده رشته از کتابخانه String-Handling
۳۴۵	۸-۸-۱ strchr تابع
۳۴۶	۸-۸-۲ strcspn تابع
۳۴۶	۸-۸-۳ strpbrk تابع
۳۴۷	۸-۸-۴ strchr تابع
۳۴۷	۸-۸-۵ strspn تابع
۳۴۸	۸-۸-۶ strstr تابع
۳۴۹	۸-۸-۷ strtok تابع



۳۵۰.....	۸-۹ String-Handling	توابع حافظه از کتابخانه
۳۵۱.....	۸-۹-۱ memcopy	تابع
۳۵۱.....	۸-۹-۲ memmove	تابع
۳۵۲.....	۸-۹-۳ memcmp	تابع
۳۵۲.....	۸-۹-۴ memchr	تابع
۳۵۳.....	۸-۹-۵ memset	تابع
۳۵۳.....	۸-۱۰ String-Handling	سایر توابع از کتابخانه
۳۵۳.....	۸-۱۰-۱ strerror	تابع
۳۵۴.....	۸-۱۰-۲ strlen	تابع
۳۵۴.....	۸-۱۱	برنامه نویسی امن در C
۳۵۶.....		خودآزمایی
۳۵۸.....		پاسخ خود آزمایی
۳۵۹.....		تمرینات
۳۶۳.....	فصل نهم: ورودی و خروجی قالب بندی شده در C	
۳۶۴.....	۹-۱	مقدمه
۳۶۴.....	۹-۲	استریم ها
۳۶۴.....	۹-۳	قالب بندی خروجی با printf
۳۶۵.....	۹-۴	چاپ مقادیر صحیح
۳۶۶.....	۹-۵	چاپ اعداد اعشاری
۳۶۷.....	۹-۵-۱	تبدیل E, e و f
۳۶۷.....	۹-۵-۲	تبدیل G و g
۳۶۷.....	۹-۵-۳	توصیف عملکرد تصریح کننده های تبدیل اعشاری
۳۶۸.....	۹-۶	چاپ رشته و کاراکتر
۳۶۹.....	۹-۷	دیگر تصریح کننده های تبدیل
۳۷۰.....	۹-۸	چاپ در پهنای فیلد و دقت
۳۷۰.....	۹-۸-۱	مشخص کردن پهنای فیلد برای چاپ اعداد صحیح
۳۷۰.....	۹-۸-۲	مشخص کردن دقت اعداد صحیح، اعشاری و رشته
۳۷۱.....	۹-۸-۳	ترکیب پهنای فیلد و دقت
۳۷۲.....	۹-۹	استفاده از پرچم ها در رشته کنترلی فرمت printf
۳۷۲.....	۹-۹-۱	تراز بندی چپ و راست
۳۷۲.....	۹-۹-۲	چاپ اعداد مثبت و منفی با و بدون پرچم +
۳۷۳.....	۹-۹-۳	استفاده از پرچم Space
۳۷۳.....	۹-۹-۴	استفاده از پرچم #
۳۷۴.....	۹-۹-۵	استفاده از پرچم 0
۳۷۴.....	۹-۱۰	چاپ لیترال و توالی escape
۳۷۵.....	۹-۱۱	خواندن ورودی قالب بندی شده با scanf



۳۷۵	۹-۱۱-۱ گرامر scanf
۳۷۵	۹-۱۱-۲ تصریح کننده‌های تبدیل scanf
۳۷۶	۹-۱۱-۳ خواندن اعداد صحیح با scanf
۳۷۶	۹-۱۱-۴ خواندن اعداد اعشاری با scanf
۳۷۷	۹-۱۱-۵ خواندن کاراکترها و رشته‌ها با scanf
۳۷۸	۹-۱۱-۶ استفاده از Scan Set با scanf
۳۷۹	۹-۱۱-۷ استفاده از پهنای فیلد با scanf
۳۷۹	۹-۱۱-۸ پرش از کاراکترها در استریم ورودی
۳۸۰	۹-۱۲ برنامه‌نویسی امن در C
۳۸۱	خودآزمایی
۳۸۳	پاسخ خودآزمایی
۳۸۴	تمرینات
۴۸۷	فصل دهم: مختار، اجتماع، دستکاری بیت و شمارشی در C
۳۸۸	۱۰-۱ مقدمه
۳۸۸	۱۰-۲ تعریف ساختار
۳۸۹	۱۰-۲-۱ ساختار خودآزمایی
۳۸۹	۱۱-۲-۲ تعریف متغیرهای مختار
۳۹۰	۱۰-۲-۳ نام ساختار
۳۹۰	۱۰-۲-۴ عملیات‌های که می‌توان بر روی شماره انجام داد
۳۹۱	۱۰-۳ مقداردهی ساختارها
۳۹۱	۱۰-۴ دسترسی به اعضا ساختار با > و
۳۹۲	۱۰-۵ استفاده از ساختارها در توابع
۳۹۳	۱۰-۶ typedef
۳۹۳	۱۰-۷ مثال: بازی کارت با کارایی بالا
۳۹۵	۱۰-۸ Union
۳۹۶	۱۰-۸-۱ اعلان اجتماع
۳۹۶	۱۰-۸-۲ عملیات‌های که می‌توان با اجتماع‌ها انجام داد
۳۹۶	۱۰-۸-۳ مقداردهی اجتماع‌ها در اعلان‌ها
۳۹۷	۱۰-۸-۴ نحوه استفاده از اجتماع‌ها
۳۹۷	۱۰-۹ عملگرهای بیتی
۳۹۹	۱۰-۹-۱ نمایش بیتی یک مقدار صحیح بدون علامت
۴۰۰	۱۰-۹-۲ ساخت تابع displayBits با کارایی بالاتر
۴۰۱	۱۰-۹-۳ استفاده از عملگرهای بیتی AND و OR فراگیر بیتی، OR بیتی انحصاری و متمم بیتی
۴۰۳	۱۰-۹-۴ استفاده از عملگرهای بیتی شیفت چپ و راست
۴۰۵	۱۰-۹-۵ عملگرهای تخصیص بیتی
۴۰۶	۱۰-۱۰ فیلد بیت



۴۰۶	۱۰-۱-۱ تعریف فیلد بیت
۴۰۷	۱۰-۱۰-۲ استفاده از فیلدهای بیت برای ارائه کارت بازی
۴۰۸	۱۰-۱۰-۳ فیلد بیتی بی نام
۴۰۹	۱۰-۱۱ ثابت های شمارشی
۴۱۰	۱۰-۱۲ ساختارها و اجتماع های ناشناس
۴۱۱	۱۰-۱۳ برنامه نویسی امن در C
۴۱۳	خودآزمایی
۴۱۵	پاسخ خود آزمایی
۴۱۶	تمرینات
۴۱۹	فصل یازدهم: پردازش فایل در C
۴۲۰	۱۱-۱ مقدمه
۴۲۰	۱۱-۲ فایل ها و استریم ها
۴۲۱	۱۱-۳ ایجاد فایل ترتیبی
۴۲۲	۱۱-۳-۱ اشاره گر به FILE
۴۲۳	۱۱-۳-۲ استفاده از fopen برای باز کردن فایل
۴۲۳	۱۱-۳-۳ استفاده از feof برای بررسی نشانه EOF
۴۲۴	۱۱-۳-۴ استفاده از fprintf برای نوشتن در فایل
۴۲۴	۱۱-۳-۵ استفاده از fclose برای بستن فایل
۴۲۵	۱۱-۳-۶ مدهای باز کردن فایل
۴۲۷	۱۱-۴ خواندن داده از یک فایل با دسترسی ترتیبی
۴۲۸	۱۱-۴-۱ تنظیم اشاره گر موقعیت فایل
۴۲۸	۱۱-۴-۲ برنامه مدیریت اعتبار
۴۳۱	۱۱-۵ فایل با دسترسی تصادفی
۴۳۲	۱۱-۶ ایجاد فایل با دسترسی تصادفی
۴۳۵	۱۱-۷ نوشتن داده بصورت تصادفی در فایل با دسترسی تصادفی
۴۳۶	۱۱-۷-۱ تنظیم موقعیت اشاره گر موقعیت فایل با fseek
۴۳۷	۱۱-۷-۲ بررسی خطا
۴۳۸	۱۱-۸ خواندن داده از فایل با دسترسی تصادفی
۴۳۹	۱۱-۹ میخت آموزشی: برنامه پردازش تراکنشی
۴۴۴	۱۱-۱۰ برنامه نویسی امن در C
۴۴۶	خودآزمایی
۴۴۷	پاسخ خودآزمایی
۴۴۸	تمرینات
۴۵۱	فصل دوازدهم: ساختمان های داده در C
۴۵۲	۱۲-۱ مقدمه
۴۵۲	۱۲-۲ ساختارهای خود ارجاعی



۴۵۳		۱۲-۳	اخذ دینامیکی حافظه
۴۵۴		۱۲-۴	لیست‌های پیوندی
۴۵۹		۱۲-۴-۱	insert تابع
۴۶۱		۱۲-۴-۲	delete تابع
۴۶۲		۱۲-۴-۳	printList تابع
۴۶۳		۱۲-۵	پشته
۴۶۶		۱۲-۵-۱	push تابع
۴۶۷		۱۲-۵-۲	pop تابع
۴۶۸		۱۲-۵-۳	کاربردهای پشته
۴۶۸		۱۲-۶	صف
۴۷۲		۱۲-۶-۱	enqueue تابع
۴۷۳		۱۲-۶-۲	dequeue تابع
۴۷۴		۱۲-۷	درخت
۴۷۷		۱۲-۷-۱	insert Node تابع
۴۷۸		۱۲-۷-۲	پیمایش توابع inOrder, preOrder و postOrder
۴۷۹		۱۲-۷-۳	حذف عناصر تک‌ارگی
۴۷۹		۱۲-۷-۴	درخت جستجوی دودویی
۴۷۹		۱۲-۸	برنامه‌نویسی امن در C
۴۸۱			خود آزمایی
۴۸۳			پاسخ خود آزمایی
۴۸۴			تمرینات
۴۹۱			فصل سیزدهم: پیش‌پردازنده در C
۴۹۲		۱۳-۱	مقدمه
۴۹۲		۱۳-۲	رهنمود پیش‌پردازنده #include
۴۹۳		۱۳-۳	رهنمود پیش‌پردازنده #define ثابت‌های سمبولیک
۴۹۳		۱۳-۴	رهنمود پیش‌پردازنده #define ماکروها
۴۹۴		۱۳-۴-۱	ماکرو با یک آرگومان
۴۹۵		۱۳-۴-۲	ماکرو با دو آرگومان
۴۹۵		۱۳-۴-۳	ماکرو با کاراکتر ادامه
۴۹۵		۱۳-۴-۴	رهنمود پیش‌پردازنده #undef
۴۹۵		۱۳-۴-۵	توابع کتابخانه استاندارد و ماکروها
۴۹۵		۱۳-۴-۶	قرار ندادن عبارت با عوارض در ماکروها
۴۹۶		۱۳-۵	کامپایل شرطی
۴۹۶		۱۳-۵-۱	رهنمود پیش‌پردازنده #if...#endif
۴۹۶		۱۳-۵-۲	عدم کامپایل بخشی از کد با #if...#endif
۴۹۶		۱۳-۵-۳	کامپایل شرطی برای دیباگ کد



۴۹۷	۱۳-۶ پیش پردازنده #error و #pragma
۴۹۷	۱۳-۷ عملگرهای # و ##
۴۹۸	۱۳-۸ شماره‌های خط
۴۹۸	۱۳-۹ ثابت‌های سمبولیک از قبل تعریف شده
۴۹۸	۱۳-۱۰ تایید کننده‌ها
۴۹۹	۱۳-۱۱ برنامه‌نویسی امن در C
۵۰۰	خود آزمایی
۵۰۱	پاسخ خود آزمایی
۵۰۲	تمرینات
۵۰۳	فصل چهاردهم: مباحث دیگر در C
۵۰۴	۱۴-۱ مقدمه
۵۰۴	۱۴-۲ هدایت I/O
۵۰۴	۱۴-۲-۱ هدایت ورودی <
۵۰۴	۱۴-۲-۲ هدایت ورودی با
۵۰۵	۱۴-۲-۳ هدایت خروجی
۵۰۵	۱۴-۳ لیست آرگومان با طول متغیر
۵۰۷	۱۴-۴ استفاده از آرگومان‌های خط فرمان
۵۰۸	۱۴-۵ کامپایل برنامه‌ها با چندین فایل منبع
۵۰۸	۱۴-۵-۱ اعلان extern برای متغیرهای سراسری در سایر فایل‌ها
۵۰۹	۱۴-۵-۲ پیش‌نمونه‌های تابع
۵۰۹	۱۴-۵-۳ محدود کردن قلمرو با static
۵۱۰	۱۴-۵-۴ Makefile
۵۱۰	۱۴-۶ خاتمه برنامه با exit و atexit
۵۱۲	۱۴-۷ پسوندهای ثابت صحیح و اعشاری
۵۱۲	۱۴-۸ رسیدگی به سیگنال
۵۱۴	۱۴-۹ اخذ دینامیکی حافظه: توابع calloc و realloc
۵۱۵	۱۴-۱۰ انشعاب غیرشرطی با goto
۵۱۷	خود آزمایی
۵۱۸	پاسخ خود آزمایی
۵۱۸	تمرینات
۵۱۹	فصل پانزدهم: ++C بهتر از C: مقدمه‌ای بر تکنولوژی شی
۵۲۰	۱۵-۱ مقدمه
۵۲۰	۱۵-۲ ++C
۵۲۰	۱۵-۳ یک برنامه ساده: جمع دو عدد صحیح
۵۲۰	۱۵-۳-۱ برنامه جمع در ++C
۵۲۱	۱۵-۳-۲ سرآیند <iostream>



۵۲۱	main تابع ۱۵-۳-۳
۵۲۲	۱۵-۳-۴ اعلان متغیر
۵۲۲	۱۵-۳-۵ استریم ورودی و خروجی استاندارد
۵۲۲	std::endl ۱۵-۳-۶
۵۲۳	std:: ۱۵-۳-۷
۵۲۳	۱۵-۳-۸ متصل کردن استریم‌های خروجی
۵۲۳	۱۵-۳-۹ عدم استفاده از عبارت return در main
۵۲۳	۱۵-۳-۱۰ سربارگذاری عملگر
۵۲۴	۱۵-۴ کتابخانه استاندارد C++
۵۲۴	۱۵-۵ قابل‌های سرآیند
۵۲۶	۱۵-۶ inline
۵۲۸	۱-۷ نامگذاری کلیدی C++
۵۲۹	۱۵-۸ مراجعه و بازگشت‌های مراجعه
۵۲۹	۱۵-۸-۱ پارامترهای مراجعه
۵۳۰	۱۵-۸-۲ ارسال آرگومان با مراجعه و با مراجعه
۵۳۱	۱۵-۸-۳ مراجعه‌ها به زبان نامی مستعار در درون یک تابع
۵۳۲	۱۵-۸-۴ برگشت دادن یک مراجعه در یک تابع
۵۳۲	۱۵-۹ لیست پارامتری تهی
۵۳۳	۱۵-۱۰ آرگومان‌های پیش‌فرض
۵۳۵	۱۵-۱۱ عملگر تفکیک قلمرو غیرباینری
۵۳۵	۱۵-۱۲ سربارگذاری تابع
۵۳۸	۱۵-۱۳ الگوهای تابع
۵۳۸	۱۵-۱۳-۱ تعریف یک الگوی تابع
۵۳۹	۱۵-۱۳-۲ استفاده از الگوی تابع
۵۴۱	۱۵-۱۴ مقدمه‌ای بر تکنولوژی شی و UML
۵۴۱	۱۵-۱۴-۱ مفاهیم پایه تکنولوژی شی
۵۴۲	۱۵-۱۴-۲ کلاس‌ها، اعضا داده و توابع عضو
۵۴۳	۱۵-۱۴-۳ معرفی Object-Oriented Analysis and Design (OOAD)
۵۴۴	۱۵-۱۴-۴ زبان UML
۵۴۵	۱۵-۱۵ کلاس الگوی vector
۵۴۵	۱۵-۱۵-۱ مشکلات مرتبط با آرایه‌های مبتنی بر اشاره‌گر سبک C
۵۴۵	۱۵-۱۵-۲ استفاده از الگوی کلاس vector
۵۵۰	۱۵-۱۵-۳ رسیدگی به استثنا: پردازش شاخص خارج از محدوده
۵۵۲	خود آزمایی
۵۵۳	پاسخ خود آزمایی
۵۵۴	تمرینات



۵۵۵	فصل شانزدهم: معرفی کلاس، شی و رشته
۵۵۶	۱۶-۱ مقدمه
۵۵۶	۱۶-۲ تعریف کلاس با تابع عضو
۵۶۰	۱۶-۳ تعریف تابع عضو با پارامتر
۵۶۳	۱۶-۴ اعضای داده، توابع get و set
۵۷۰	۱۶-۵ مقداردهی اولیه شی‌ها با سازنده‌ها
۵۷۸	۱۶-۷ جداسازی واسط از پیاده‌سازی
۵۸۴	۱۶-۸ اعتبارسنجی داده با توابع set
۵۹۰	خودآزمایی
۵۹۱	پاسخ خودآزمایی
۵۹۱	تمرینات
۵۹۳	فصل هفدهم: کلاس‌ها؛ گام‌های اولیه؛ راه‌اندازی استثناء
۵۹۴	۱۷-۱ مقدمه
۵۹۵	۱۷-۲ مبحث آموزشی: کلاس Time
۶۰۱	۱۷-۳ قلمرو کلاس و دسترسی به اعضا کلاس
۶۰۲	۱۷-۴ توابع دسترسی و توابع یوتیلیتی
۶۰۲	۱۷-۵ مبحث آموزشی کلاس Time: سازنده‌ها، آرگومان‌های پیش فرض
۶۰۸	۱۷-۶ نابود کننده‌ها
۶۰۹	۱۷-۷ زمان فراخوانی سازنده‌ها و نابود کننده‌ها
۶۱۲	۱۷-۸ مبحث آموزشی کلاس Time: برگشت دادن یک مراجعه به داده عضو private
۶۱۴	۱۷-۹ تخصیص memberwise
۶۱۶	۱۷-۱۰ شی‌های ثابت و توابع عضو ثابت
۶۱۸	۱۷-۱۱ ترکیب: شی‌ها به عنوان اعضای کلاس
۶۲۳	۱۷-۱۲ توابع و کلاس‌های friend
۶۲۶	۱۷-۱۳ استفاده از اشاره‌گر this
۶۳۱	۱۷-۱۴ اعضای استاتیک کلاس
۶۳۶	خودآزمایی
۶۳۷	پاسخ خودآزمایی
۶۳۷	تمرینات
۶۴۱	فصل هیجدهم: سربارگذاری عملگر؛ کلاس string
۶۴۲	۱۸-۱ مقدمه
۶۴۲	۱۸-۲ استفاده از عملگرهای سربارگذاری شده کلاس string از کتابخانه استاندارد
۶۴۶	۱۸-۳ اصول سربارگذاری عملگر
۶۴۸	۱۸-۴ سربارگذاری عملگرهای باینری
۶۴۸	۱۸-۵ سربارگذاری عملگرهای درج و استخراج
۶۵۱	۱۸-۶ سربارگذاری عملگرهای غیرباینری



۶۵۲	۱۸-۷ سربارگذاری ++ و --
۶۵۳	۱۸-۸ بحث آموزشی: کلاس Date
۶۵۸	۱۸-۹ مدیریت دینامیکی حافظه
۶۶۰	۱۸-۱۰ بحث آموزشی: کلاس Array
۶۶۱	۱۸-۱۰-۱ استفاده از کلاس Array
۶۶۵	۱۸-۱۰-۲ تعریف کلاس Array
۶۷۴	۱۸-۱۱ عملگر به عنوان عضو در برابر تابع غیر عضو
۶۷۴	۱۸-۱۲ تبدیل مابین نوع‌ها
۶۷۶	۱۸-۱۳ سازنده‌های explicit و عملگرهای تبدیل
۶۷۸	۱۸-۱۴ عملگر فراخوانی تابع () سربارگذاری شده
۶۷۹	خود آزمایی
۶۸۰	پاسخ خود آزمایی
۶۸۰	تمرینات
۶۸۷	فصل نوزدهم: برنامه نویسی شی‌گرا: توارث
۶۸۸	۱۹-۱ مقدمه
۶۸۹	۱۹-۲ کلاس‌های مبنا و کلاس‌های مشتق شده
۶۹۲	۱۹-۳ ارتباط مابین کلاس‌های مبنا و کلاس‌های مشتق شده
۶۹۲	۱۹-۳-۱ ایجاد و استفاده از کلاس CommissionEmployee
۶۹۷	۱۹-۳-۲ ایجاد کلاس BasePlusCommissionEmployee بدون استفاده از توارث
۷۰۳	۱۹-۳-۳ ایجاد سلسله مراتب توارث CommissionEmployee-BasePlusCommissionEmployee
	۱۹-۳-۴ سلسله مراتب توارث CommissionEmployee-BasePlusCommissionEmployee با
۷۰۷	استفاده از داده protected
	۱۹-۳-۵ سلسله مراتب توارث CommissionEmployee-BasePlusCommissionEmployee با
۷۱۰	استفاده از داده private
۷۱۵	۱۹-۴ سازنده‌ها و پایان‌دهنده‌ها در کلاس‌های مشتق شده
۷۱۷	۱۹-۵ توارث protected, public و private
۷۱۸	۱۹-۶ مهندسی نرم‌افزار به کمک توارث
۷۱۹	خود آزمایی
۷۲۰	پاسخ خود آزمایی
۷۲۰	تمرینات
۷۲۱	فصل بیستم: برنامه نویسی شی‌گرا: چندریختی
۷۲۲	۲۰-۱ مقدمه
۷۲۳	۲۰-۲ چند مثال از چندریختی
۷۲۵	۲۰-۳ رابطه مابین شی‌ها در سلسله مراتب توارث
۷۲۶	۲۰-۳-۱ فراخوانی توابع کلاس مبنا از طریق شی‌های کلاس مشتق شده
۷۲۹	۲۰-۳-۲ هدایت اشاره‌گرهای کلاس مشتق شده به طرف شی‌های کلاس مشتق شده



۷۳۰.....	۲۰-۳-۳ فراخوانی تابع عضو کلاس مشتق شده از طریق اشاره‌گرهای کلاس مینا
۷۳۱.....	۲۰-۳-۴ توابع و نابود کننده‌های virtual
۷۳۸.....	۲۰-۴ نوع فیلدها و عبارات switch
۷۳۹.....	۲۰-۵ کلاس‌های انتزاعی و توابع virtual محض
۷۴۱.....	۲۰-۶ مبحث آموزشی: سیستم پرداخت حقوق با استفاده از چند ریختی
۷۴۳.....	۲۰-۶-۱ ایجاد کلاس مبنای انتزاعی Employee
۷۴۶.....	۲۰-۶-۲ SalariedEmployee ایجاد کلاس مشتق شده مقید
۷۴۸.....	۲۰-۶-۳ CommissionEmployee ایجاد کلاس مشتق شده مقید
۷۵۶.....	۲۰-۷ چند ریختی توابع virtual و مقیدسازی دینامیکی
	۲۰-۸ مبحث آموزشی: سیستم پرداخت حقوق با استفاده از چند ریختی و اطلاعات نوع زمان اجرا با
۷۶۰.....	تبدیل dynamic_cast, typeid و type_info
۷۶۴.....	خودآزمایی
۷۶۵.....	پاسخ خودآزمایی
۷۶۵.....	تمرینات
۷۶۷.....	فصل بیست و یکم: استریم ورودی/خروجی
۷۶۸.....	۲۱-۱ مقدمه
۷۶۸.....	۲۱-۲ استریم‌ها
۷۶۹.....	۲۱-۲-۱ استریم‌های کلاسیک در مقابل استریم‌های استاندارد
۷۷۰.....	۲۱-۲-۲ سرآیندهای کتابخانه iostream
۷۷۰.....	۲۱-۲-۳ کلاس و شی‌های استریم ورودی/خروجی
۷۷۳.....	۲۱-۳ استریم خروجی
۷۷۳.....	۲۱-۳-۱ چاپ متغیرهای char *
۷۷۳.....	۲۱-۳-۲ چاپ کاراکتر با استفاده از تابع عضو put
۷۷۴.....	۲۱-۴ استریم ورودی
۷۷۴.....	۲۱-۴-۱ توابع عضو get و getline
۷۷۷.....	۲۱-۴-۲ توابع عضو peek, putback و ignore
۷۷۷.....	۲۱-۴-۳ I/O نوع ایمن
۷۷۷.....	۲۱-۵ ورودی/خروجی قالب‌بندی نشده با استفاده از read, write و gcout
۷۷۸.....	۲۱-۶ معرفی دستکاری کننده‌های استریم
۷۷۹.....	۲۱-۶-۱ پایهٔ انتگرال استریم: dec, oct, hex و setbase
۷۷۹.....	۲۱-۶-۲ دقت نقطه اعشار (precision, setprecision)
۷۸۱.....	۲۱-۶-۳ طول میدان (width و setw)
۷۸۱.....	۲۱-۶-۴ دستکاری کننده‌های استریم خروجی تعریف شده توسط کاربر
۷۸۳.....	۲۱-۷ تعیین فرمت استریم و دستکاری کننده‌های استریم
۷۸۳.....	۲۱-۷-۱ دنباله صفرها و نقاط دسیمال (showpoint)
۷۸۴.....	۲۱-۷-۲ ترازبندی (left, right و internal)



۷۸۵.....	۲۱-۷-۳ لایه گذاری (setfill, fill).....
۷۸۶.....	۲۱-۷-۴ پایه انتگرال استریم (showbase, hex, oct, dec).....
۷۸۷.....	۲۱-۷-۵ اعداد اعشاری، نماد علمی و ثابت (fixed, scientific).....
۷۸۸.....	۲۱-۷-۶ کنترل حروف بزرگ/کوچک (uppercase).....
۷۸۹.....	۲۱-۷-۷ قالب بندی بولی (boolalpha).....
۷۹۰.....	۲۱-۷-۸ تنظیم و تنظیم مجدد وضعیت قالب بندی از طریق تابع عضو flags.....
۷۹۱.....	۲۱-۸ وضعیت خطا در استریم.....
۷۹۳.....	۲۱-۹ پیوند استریم خروجی با استریم ورودی.....
۷۹۴.....	تمرینات.....
۷۹۵.....	فصل بیست و نهم: رسیدگی به استثنا
۷۹۶.....	۲۲-۱ مقدمه.....
۷۹۶.....	۲۲-۲ مثال: تقسیم بر صفر.....
۸۰۲.....	۲۲-۳ راه اندازی استثنا.....
۸۰۳.....	۲۲-۴ باز کردن بسته.....
۸۰۵.....	۲۲-۵ زمان استفاده از رسیدگی به استثنا.....
۸۰۶.....	۲۲-۶ سازنده ها، نابود کننده ها و رسیدگی به استثنا.....
۸۰۶.....	۲۲-۷ استثناها و توارث.....
۸۰۷.....	۲۲-۸ پردازش واماندگی new.....
۸۰۹.....	۲۲-۹ کلاس unique_ptr و اخذ حافظه پینامیک.....
۸۱۲.....	۲۲-۱۰ کتابخانه سلسله مراتب استاندارد استثنا.....
۸۱۵.....	فصل بیست و سوم: الگوها
۸۱۶.....	۲۳-۱ مقدمه.....
۸۱۶.....	۲۳-۲ الگوهای کلاس.....
۸۲۰.....	۲۳-۳ الگوی تابع برای دست کاری یک شی از الگوی کلاس.....
۸۲۲.....	۲۳-۴ پارامترهای بدون نوع.....
۸۲۲.....	۲۳-۵ آرگومان های پیش فرض برای پارامترهای نوع الگو.....
۸۲۳.....	۲۳-۶ سربار گذاری الگوهای تابع.....
۸۲۴.....	تمرینات.....