

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

سرشناسه	: تراپی، فرشاد، ۱۳۵۴ -
عنوان و نام پدیدآور	: برنامه‌نویسی شی‌گرا به زبان ++C / فرشاد تراپی
مشخصات نشر	: تهران: دانشگاه صنعتی خواجه نصیرالدین طوسی، ۱۳۹۱.
مشخصات ظاهری	: ۳۷۲ ص.: مصور، جدول، نمودار
شابک	: ۱۲۵۰۰۰ ریال ۸-۳۷-۶۳۸۳-۶۰۰-۹۷۸:
وضعیت فهرست نویسی	: فیپا
یادداشت	: واژه‌نامه
یادداشت	: کتابنامه.
موضوع	: برنامه‌نویسی شی‌گرا
موضوع	: سی ++ (زبان برنامه‌نویسی کامپیوتر)
شناسه افزوده	: دانشگاه صنعتی خواجه نصیرالدین طوسی
رده بندی کنگره	: ۱۳۹۱ ب۳۶ ت / ۶۴ / QAV۶
رده بندی دیویی	: ۰۰۵/۱۱۷
شماره کتابشناسی ملی	: ۳۰۹۳۵۵۳

نام کتاب: برنامه‌نویسی شی‌گرا به زبان ++C

مولف: دکتر فرشاد تراپی، عضو هیأت علمی دانشکده مهندسی مکانیک، دانشگاه صنعتی خواجه نصیرالدین طوسی

ناشر: دانشگاه صنعتی خواجه نصیرالدین طوسی

نوبت چاپ: اول

تاریخ چاپ: اسفندماه ۱۳۹۱

تیراژ: ۱۰۰۰ جلد

کد کتاب: ۳۲۳

قیمت: ۱۲۵۰۰ تومان

ISBN: 978-600-6383-37-8

شابک: ۹۷۸-۶۰۰-۶۳۸۳-۳۷-۸

ویراستار و صفحه آرا: وحید دامن‌افشان

چاپ و لیتوگرافی: هوررنگ

صحافی: گرنامی

آدرس و تلفن مرکز پخش و فروش: خیابان ولیعصر (عج)، بالاتر از میدان ونک، تقاطع میرداماد،

روبروی ساختمان اسکان (۰۲۱-۸۸۷۷۲۲۷۷)

(حق چاپ برای ناشر محفوظ است)

# برنامه‌نویسی شی‌گرا به زبان C++

دکتر فرشاد ترابی

عضو هیأت علمی دانشگاه خواجه نصیرالدین طوسی

انتشارات دانشگاه خواجه نصیرالدین طوسی

۱۳۹۱



تقدیم به همسر عزیزم، شیوا

پدر و مادر مهربانم

ویگانه فرزندم، آریا



## پیش‌گفتار

در دنیای امروز، نرم‌افزارهای رایانه‌ای نقش بسیار مهمی را در زندگی بشر ایفا کرده به طوری که با یک نگاه ساده به اطراف خود می‌توان انبوه برنامه‌ها و نرم‌افزارهای رایانه‌ای را مشاهده کرد. تمامی برنامه‌های موجود در گوشی‌های همراه، رایانه‌های شخصی، سیستم‌های کنترلی، سیستم‌های مخابراتی، لوازم صوتی و تصویری، خودروها و بسیاری موارد دیگر تنها بخشی از مثال‌های موجود هستند. نقش برنامه‌های رایانه‌ای در پیشرفت تکنولوژی آن قدر زیاد است که بدون اغراق باید گفت که پیشرفت تکنولوژی بدون وجود برنامه‌ها و نرم‌افزارهای رایانه‌ای امکان‌پذیر نیست. از طرفی، تنوع و گستره‌ی تکنولوژی باعث شده که علم برنامه‌نویسی به یک علم کاملاً بزرگ و پیچیده تبدیل شده و شاخه‌های بسیار زیادی را شامل شود. برای نیل به اهداف مختلف، زبان‌های برنامه‌نویسی گوناگونی به وجود آمده‌اند. بسیاری از زبان‌های مختلف برنامه‌نویسی عمر چندان زیادی نداشتند و تنها برخی از آنها توانسته‌اند نیازهای بشر در تولید نرم‌افزار را برآورده کنند. در این میان، زبان برنامه‌نویسی C/C++، یکی از معروف‌ترین و پرکاربردترین زبان‌های موجود در دنیا بوده که بسیاری از نرم‌افزارهای کنونی، به وسیله‌ی آن نوشته شده است. وجود خواص و ابزارهایی نظیر سطح میانی، سرعت پردازش بالا، قابلیت برنامه‌نویسی شی‌گرا، کار با سخت‌افزار و موارد دیگر را می‌توان از جمله جذابیت‌های این زبان دانست. این خاصیت‌ها و توانایی‌ها باعث شده تا با وجود قدمت، این زبان همچنان جزو زبان‌های قدرتمند و قابل رقابت بین زبان‌های مختلف دنیا باشد.

زبان‌های مختلف برنامه‌نویسی بنا به نیازهای مختلف به‌وجود آمده و می‌آیند. به‌عنوان مثال، زبان بیسیک به‌عنوان یک زبان کاملاً ابتدایی و ساده مطرح بوده که برای مواردی که قرار است برنامه‌های بسیار ساده نوشته شود، به‌وجود آمده است. زبان فرترن، زبانی است که برای انجام محاسبات عددی سنگین و طولانی طراحی شده است. این زبان قابلیت استفاده از حجم زیادی از حافظه را داشته و سرعت انجام محاسبات آن بسیار زیاد است. همچنین یادگیری آن، نسبت به برخی از زبان‌های دیگر، ساده‌تر است. به همین دلیل این زبان برای برنامه‌نویسان رشته‌های مختلف مهندسی به غیر از مهندسی رایانه، بسیار مناسب بوده و آموزش آن در این رشته‌ها رواج زیادی دارد. زبان برنامه‌نویسی C/C++ زبانی است که برای انجام محاسبات عددی، کارهای گرافیکی، ارتباط با سخت‌افزار و غیره طراحی شده است. این زبان از لحاظ انعطاف‌پذیری و کارایی، زبانی بسیار مفید بوده اما به‌علت آنکه قابلیت‌های آن بسیار زیاد است، بیشتر مورد توجه مهندسان علوم رایانه قرار گرفته و در علوم دیگر کمتر آموزش داده می‌شود. به همین علت، در اغلب رشته‌های مهندسی، این زبان کمتر آموزش داده شده در حالی که تجربه‌ی برنامه‌نویسی شخصی حاکی از آن است که چنانچه اصول آن به‌طور مناسب آموزش داده شود، دانشجویان رشته‌های دیگر نیز قادر خواهند بود از توانایی‌های آن در انجام امور مربوط به خود استفاده کرده و از ابزارهای مختلف آن در برنامه‌نویسی استفاده کنند.

آنچه که در آموختن یک زبان برنامه‌نویسی اهمیت دارد، توجه به اساس و اصول برنامه‌نویسی است و انتخاب زبان از اهمیت کمتری برخوردار است. به عبارت دیگر، اصول برنامه‌نویسی مستقل از زبان بوده و چنانچه یک برنامه‌نویس این اصول و قوانین را یاد بگیرد، با هر زبانی که به وی پیشنهاد شود، به راحتی می‌تواند ارتباط برقرار کند. اگر چه تنوع زبان‌های برنامه‌نویسی ممکن است شخص را در انتخاب دچار مشکل کند، اما با آموزش یک یا دو زبان مختلف در خواهد یافت که تمامی آن‌ها از یک سری اصول مشترک پیروی می‌کنند. بنابراین به زبان‌آموزان همواره توصیه می‌شود که به‌جای انتخاب بین زبان‌های مختلف، تمرکز خود را روی اصول و قوانین پایه‌ای برنامه‌نویسی قرار دهند.

تجربه‌ی شخصی اینجانب در نگارش کدهای مهندسی با زبان‌های گوناگون نشان داد که زبان برنامه‌نویسی C/C++ یکی از بهترین گزینه‌های موجود برای نوشتن کدهای مهندسی است. وجود ابزارهایی نظیر اشاره‌گر و قابلیت برنامه‌نویسی شی‌گرا در این زبان باعث می‌شود تا کدهای مهندسی با سرعت و دقت بسیار بالاتری نسبت به زبان‌های دیگر قابل توسعه و بسط باشد. اما باید دقت کرد که درک این مفاهیم برای مهندسان غیر رایانه‌ای کمی سخت بوده و آموزش آن



مستلزم تلاش و صرف وقت است. همین موضوع، انگیزه‌ی اصلی برای نگارش این کتاب را به‌وجود آورد. نگارش این کتاب به‌گونه‌ای است که در آن هدف اصلی، آموزش زبان برنامه‌نویسی C/C++ برای مهندسانی است که اصولاً با این مفاهیم آشنایی نداشته‌اند. تمامی مطالب بیان شده در این کتاب به‌صورت قدم به قدم و با ذکر انواع مثال‌های مختلف تنظیم شده است به‌گونه‌ای که زبان‌آموزان بتوانند به‌راحتی با این زبان مهم ارتباط برقرار کنند. سابقه‌ی تدریس این کتاب در طول سال‌های مختلف در دانشکده‌ی مهندسی مکانیک در دانشگاه صنعتی خواجه نصیرالدین طوسی نشان داد که روند آموزش این کتاب بسیار مناسب بوده و دانشجویان این رشته بعد از گذشت یک ترم، آشنایی کاملاً مناسبی از این زبان کسب کرده‌اند. البته باید ذکر کرد که تمرکز اصلی این کتاب روی آموزش اصول اولیه و پایه‌ی برنامه‌نویسی بوده به‌گونه‌ای که با یادگیری این کتاب، زبان‌آموزان مفاهیم پایه را کاملاً درک کرده و چنانچه برای یادگیری زبان دیگری نیز اقدام کنند، بدون مشکل قادر به آموختن آن خواهند بود.

یکی از نکاتی که در این کتاب برای اولین بار به منظور آموزش زبان طراحی شده، مدلی است به نام مدل حافظه که توسط آن روند تغییرات حافظه در رایانه حین انجام کارهای مختلف در برنامه نمایش داده شده است. اگر چه نویسنده آگاهی کامل دارد که این مدل ابتکاری یک مدل واقعی نیست، اما ابداع آن باعث شده تا مفاهیم پایه در زبان برنامه‌نویسی به طرز کاملاً مطلوبی در ذهن زبان‌آموزان شکل بگیرد. تجربه‌ی تدریس این کتاب در طی سال‌های مختلف و جمع‌آوری نظرات زبان‌آموزان که اکثراً دانشجویان رشته‌ی مهندسی مکانیک بوده‌اند، نشان داد که این مدل یکی از نقاط قوت آموزشی این کتاب بوده و بنابراین از خوانندگان عزیز درخواست می‌شود روی یادگیری این مدل، توجه ویژه‌ای داشته باشند.

بدیهی است که یکی از مواردی که باعث افزایش قدرت یادگیری یک زبان برنامه‌نویسی شده و جزو نقاط قوت کتاب‌های آموزش محسوب می‌شود، وجود مسائل حل شده و نمونه‌های عملی اجرایی است. نظر به این مهم، مجموعه‌ی کاملی از نمونه مسائل کاربردی تهیه و فراهم شده است. در این مجموعه، مسائل مختلفی با استفاده از مطالب بیان شده در کتاب حاضر حل شده است. بسیاری از مسائل موجود در نمونه‌های آموزشی، مثال‌هایی کاربردی هستند که توسط دانشجویان اینجانب ارائه شده‌اند که با توجه به سطح بالای برنامه‌نویسی به‌کار رفته در آن‌ها و همچنین میزان کاربردی بودن‌شان، دسته‌بندی و ارائه شده‌اند. با توجه به حجم بالای مجموعه‌ی مسائل کاربردی، ترجیح داده شد تا کل آن‌ها در یک جلد جداگانه انتشار یابد. بدیهی است که مجموعه‌ی مذکور در کنار آموزش‌های این کتاب می‌تواند مجموعه‌ی خوبی را جهت یادگیری

مبانی پایه‌ای و همچنین ساختار زبان برنامه‌نویسی C/C++ فراهم کند. در پایان جا دارد تا از زحمات دوست بسیار عزیزم، آقای دکتر علی رضوانی که اینجانب را با دنیای شی‌گرایی و زبان C/C++ آشنا کردند، تشکر ویژه‌ای داشته باشم. همچنین از دانشجویان دانشکده‌ی مهندسی مکانیک در دانشگاه صنعتی خواجه نصیرالدین طوسی که در طی این سال‌ها با ارائه‌ی نقطه‌نظرات مثبت و سازنده‌ی خود کمک بزرگی به طراحی و تکمیل این کتاب کردند، تشکر کرده و برای آن‌ها سلامتی و پیشرفت در زندگی را آرزو می‌کنم. در حروف‌چینی این کتاب از بسته‌ی حروف‌چینی رایگان و متن‌باز X<sub>Y</sub>Persian استفاده شده است که تماماً مدیون زحمات بی‌شمار آقای دکتر وفا خلیقی، عضو هیات علمی دانشگاه مک کواری استرالیا بوده است. ویرایش و صفحه‌آرایی این کتاب نیز توسط آقای وحید دامن‌افشان، مدرس ریاضی دانشگاه صنعتی کرمانشاه و چاپ آن با زحمات آقای مهندس محسن مصیب‌نژاد، از دانشجویان سابق اینجانب به انجام رسیده است که بدین وسیله از زحمات تمامی این دوستان قدردانی می‌کنم.

فرشاد ترابی

تهران، زمستان ۱۳۹۱

# فهرست مطالب

چ	پیش‌گفتار
ط	فهرست شکل‌ها
ف	فهرست جدول‌ها
ک	فهرست برنامه‌ها
۱	بخش اول دستوره‌ای مقدماتی زبان C/C++
۳	۱ مقدمه
۵	۱.۱ زبان‌های برنامه‌نویسی
۶	۲.۱ سیستم عامل و ارتباط با نرم‌افزار
۷	۳.۱ برنامه‌نویسی شی‌گرا
۸	۴.۱ تاریخچه‌ی زبان برنامه‌نویسی C/C++
۸	۵.۱ خلاصه
۹	مراجع

۱۱	۲	مقدمه‌ای بر زبان برنامه‌نویسی C/C++
۱۲	۱.۲	ساختار نوشتاری
۱۳	۲.۲	کلمات کلیدی
۱۳	۳.۲	سوئیچ‌های کامپایلر
۱۳	۴.۲	عملگرهای ریاضی
۱۵	۱.۴.۲	تقسیم در فضای اعداد صحیح و اعداد حقیقی
۱۶	۵.۲	تقدم و تأخر در انجام عملگرهای ریاضی
۱۷	۶.۲	فرآیند اجرای یک برنامه
۱۸	۷.۲	خطاهای تولید شده در برنامه‌نویسی
۱۹	۸.۲	خلاصه
۱۹		مراجع
۲۰		مسائل
۲۱	۳	متغیرها و ثابت‌های برنامه‌نویسی
۲۲	۱.۳	تعریف متغیرها
۲۶	۱.۱.۳	قوانین نام‌گذاری متغیرها
۲۷	۲.۱.۳	نحوهی اختصاص حافظه به یک متغیر
۲۸	۳.۱.۳	محدودهی اعتبار متغیرها
۲۹	۲.۳	ثابت‌ها و نحوهی تعریف آن‌ها
۲۹	۱.۲.۳	ثابت‌های عددی
۳۰	۲.۲.۳	ثابت‌های برجسی
۳۱	۳.۲.۳	ثابت‌های ماکروبی
۳۲	۳.۳	تعریف آرایه‌های استاتیکی
۳۴	۴.۳	تعریف یک رشته‌ی متنی
۳۶	۵.۳	تعریف متغیرهای ساختاریافته
۴۱	۶.۳	خلاصه
۴۲		مراجع
۴۳		مسائل

۴۷	دستورهای ورودی/خروجی	۴
۴۸	کتابخانه‌ی <stdio.h>	۱.۴
۴۸	دستور نوشتن روی خروجی استاندارد یا نمایشگر	۱.۱.۴
۵۲	[flags]	۱.۱.۱.۴
۵۳	[width]	۲.۱.۱.۴
۵۳	[precision]	۳.۱.۱.۴
۵۴	[type]	۴.۱.۱.۴
۵۶	دستور خواندن از ورودی استاندارد یا صفحه‌کلید	۲.۱.۴
۵۷	خواندن و نوشتن از طریق فایل‌های متنی	۳.۱.۴
۶۰	خواندن و نوشتن با استفاده از <stream>	۲.۴
۶۰	خواندن و نوشتن با درگاه‌های استاندارد	۱.۲.۴
۶۲	خواندن و نوشتن به وسیله‌ی فایل	۲.۲.۴
۶۳	خلاصه	۳.۴
۶۴	مراجع	
۶۴	مسائل	
۶۷	دستورهای کنترلی	۵
۶۷	دستور کنترل شرط، if	۱.۵
۷۰	عملگرهای منطقی	۲.۵
۷۳	دستور پرش، switch	۳.۵
۷۶	دستور حلقه‌ی تکرار با تعداد تکرار مشخص، for	۴.۵
۸۰	دستور حلقه‌ی مشروط، while	۵.۵
۸۱	خلاصه	۶.۵
۸۳	مراجع	
۸۳	مسائل	
۸۹	اشاره‌گرها و کاربرد آن در برنامه‌نویسی	۶
۹۱	نحوه‌ی تخصیص حافظه توسط سیستم عامل	۱.۶
۹۲	چگونگی دسترسی به آدرس یک متغیر	۲.۶

۹۴	.....	۳.۶	اشاره‌گر و نحوه‌ی تعریف آن
۹۶	.....	۴.۶	آشنایی بیشتر با اشاره‌گر
۹۸	.....	۵.۶	روش‌های مقداردهی یک متغیر از نوع اشاره‌گر
۹۸	.....	۱.۵.۶	استفاده از آدرس متغیرهای موجود
۱۰۰	.....	۲.۵.۶	استفاده از کلمه‌ی کلیدی new
۱۰۸	.....	۶.۶	نشستی حافظه و روش جلوگیری از آن
۱۱۰	.....	۷.۶	اشاره‌گر به آرایه‌های یک‌بعدی
۱۱۴	.....	۱.۷.۶	اشاره‌کردن به تک‌تک عناصر یک آرایه
۱۱۵	.....	۲.۷.۶	اشاره‌کردن به کل عناصر یک آرایه
۱۱۸	.....	۸.۶	اشاره‌گر به یک آرایه چندبعدی
۱۲۱	.....	۹.۶	اشاره‌گر به یک struct
۱۲۵	.....	۱۰.۶	خلاصه
۱۲۶	.....		مراجع
۱۲۶	.....		مسائل
۱۲۹		۷	<b>آرایه‌های استاتیکی و دینامیکی</b>
۱۳۱	.....	۱.۷	آرایه‌های دینامیکی یک‌بعدی
۱۳۲	.....	۲.۷	پاک‌کردن آرایه‌های دینامیکی یک‌بعدی از حافظه
۱۳۴	.....	۳.۷	اضافه‌کردن چند عنصر جدید به یک آرایه‌ی دینامیکی
۱۳۸	.....	۴.۷	آرایه‌های دینامیکی دوبعدی
۱۳۹	.....	۱.۴.۷	آرایه‌ی دینامیکی دوبعدی به‌صورت پراکنده در حافظه
۱۴۲	.....	۲.۴.۷	آرایه‌ی دینامیکی دوبعدی به‌صورت یکپارچه در حافظه
۱۴۵	.....	۵.۷	آرایه‌های دینامیکی چندبعدی
۱۴۷	.....	۶.۷	خلاصه
۱۴۸	.....		مراجع
۱۴۹	.....		مسائل
۱۵۱		۸	<b>توابع</b>
۱۵۲	.....	۱.۸	فرمت کلی یک تابع

س فهرست مطالب

۱۵۵	اعلام و تعریف یک تابع
۱۵۷	استفاده از متغیر، آدرس متغیر و اشاره گر در آرگومان تابع
۱۶۴	انتخاب روش مناسب برای ارسال متغیرها به تابع
۱۶۶	مفهوم چندریختی در توابع
۱۶۹	شکل های مختلف چندریختی در توابع
۱۷۱	خلاصه
۱۷۳	مراجع
۱۷۳	مسائل

۹ کتابخانه های استاندارد

۱۷۵	
۱۷۶	کتابخانه ی <code>&lt;math.h&gt;</code> و توابع ریاضی
۱۷۶	توابع ریاضی قبل از استاندارد C99
۱۷۸	توابع ریاضی بعد از استاندارد C99
۱۷۹	کتابخانه ی <code>&lt;string&gt;</code> و کار با رشته های متنی
۱۸۵	خلاصه
۱۸۵	مراجع
۱۸۶	مسائل

بخش دوم برنامه نویسی شی گرا

۱۸۹	
۱۹۱	آشنایی با شی و برنامه نویسی شی گرا
۱۹۲	فضاهای برنامه نویسی
۱۹۳	مبانی دیدگاه برنامه نویسی شی گرا
۱۹۴	تعریف شی
۱۹۵	تفاوت بین کلاس و شی
۱۹۷	فازبندی های اصلی برای طراحی یک برنامه ی شی گرا
۲۰۰	خلاصه
۲۰۱	مراجع

۲۰۱	مسائل
۲۰۳	<b>۱۱ کلاس و نحوه‌ی تعریف آن</b>
۲۰۴	۱.۱۱ تعریف کلاس
۲۰۹	۲.۱۱ نحوه‌ی تخصیص حافظه هنگام ساخته‌شدن یک شی
۲۱۰	۳.۱۱ استفاده از اشاره‌گر به کلاس
۲۱۲	۴.۱۱ ارسال اشیا به توابع از طریق آرگومان
۲۱۲	۵.۱۱ خلاصه
۲۱۳	مراجع
۲۱۳	مسائل
۲۱۷	<b>۱۲ وراثت</b>
۲۱۸	۱.۱۲ پدر و فرزند
۲۲۴	۲.۱۲ تخصیص حافظه در وراثت
۲۲۵	۳.۱۲ وراثت‌های متوالی
۲۲۹	۱.۳.۱۲ مفهوم میدان دید یا Scope
۲۲۹	۴.۱۲ استفاده از اشاره‌گر در کلاس‌های ارث‌بری شده
۲۳۲	۵.۱۲ قالب‌بندی کلاس‌های ارث‌بری شده
۲۳۳	۶.۱۲ بازنویسی رفتارهای ارث‌بری شده و استفاده از کلمه‌ی کلیدی virtual
۲۳۶	۷.۱۲ خلاصه
۲۳۷	مراجع
۲۳۸	مسائل
۲۴۱	<b>۱۳ کلاس‌های انتزاعی</b>
۲۴۲	۱.۱۳ نمودار UML و نحوه‌ی تعریف یک کلاس انتزاعی
۲۴۴	۲.۱۳ رفتارهای انتزاعی
۲۴۵	۱.۲.۱۳ بازنویسی رفتارهای انتزاعی
۲۴۷	۳.۱۳ اشاره‌گر به کلاس‌های انتزاعی
۲۴۷	۴.۱۳ یک مثال کاربردی



۲۵۵	۵.۱۳ خلاصه
۲۵۶	مراجع
۲۵۶	مسائل
۲۵۷	<b>۱۴ حفاظت از داده‌ها و تعیین سطوح دسترسی</b>
۲۵۸	۱.۱۴ سطوح دسترسی
۲۵۸	۱.۱.۱۴ سطح دسترسی عمومی
۲۵۹	۲.۱.۱۴ سطح دسترسی خصوصی
۲۶۱	۳.۱.۱۴ سطح دسترسی حفاظت‌شده
۲۶۱	۲.۱۴ نحوه‌ی دسترسی به متغیرهای خصوصی یک کلاس
۲۶۳	۳.۱۴ سطوح حفاظت و دسترسی به داده‌ها در ارث‌بری
۲۶۵	۴.۱۴ خلاصه
۲۶۶	مراجع
۲۶۶	مسائل
۲۶۷	<b>۱۵ سربارگذاری عملگرها</b>
۲۷۱	۱.۱۵ مفهوم سربارگذاری
۲۷۲	۲.۱۵ نحوه‌ی سربارگذاری عملگرها
۲۷۳	۱.۲.۱۵ سربارگذاری عملگرهای یک‌طرفی
۲۷۴	۲.۲.۱۵ سربارگذاری عملگرهای دوطرفی
۲۷۹	۳.۱۵ عملگر تساوی به‌صورت پیش‌فرض
۲۷۹	۴.۱۵ خلاصه
۲۸۰	مراجع
۲۸۱	مسائل
۲۸۳	<b>۱۶ سازنده‌ها و مخرب‌ها</b>
۲۸۷	۱.۱۶ تعریف سازنده
۲۸۹	۲.۱۶ سازنده‌ی پیش‌فرض و سازنده‌های دیگر
۲۹۱	۳.۱۶ مخرب

۲۹۴	.....	۴.۱۶ خلاصه
۲۹۵	.....	مراجع
۲۹۵	.....	مسائل
۲۹۷		آ نحوه‌ی ذخیره‌سازی اعداد شناور در حافظه
۲۹۸	.....	۱.آ تبدیل یک عدد به مانتیس و توان
۲۹۸	.....	۲.آ ذخیره‌سازی اعداد float
۲۹۹	.....	۳.آ ذخیره‌سازی اعداد double
۳۰۱		ب ارسال تابع به‌عنوان آرگومان تابع دیگر
۳۰۵		پ کدهای ASCII
۳۱۵		واژه‌نامه انگلیسی-فارسی
۳۲۳		واژه‌نامه فارسی-انگلیسی

# فهرست شکل‌ها

۵	نمایش ساده‌ی ورود، پردازش و نمایش اطلاعات	۱.۱
۶	نمایش سطوح مختلف زبان‌های برنامه‌نویسی	۲.۱
۲۸	مدل حافظه‌ی رایانه برای اختصاص حافظه به متغیرهای برنامه‌ی ۱.۳	۱.۳
۲۸	نمایش حافظه‌ی اختصاص‌یافته به متغیر <code>z</code> بعد از مقداردهی شدن	۲.۳
۳۴	نمایش حافظه‌ی اختصاص‌یافته برای آرایه‌ی حاصل از برنامه‌ی ۶.۳	۳.۳
۴۰	نمایش حافظه‌ی اختصاص‌یافته به متغیر <code>Ali</code> و <code>studentList</code> حاصل از برنامه‌ی ۱۲.۳	۴.۳
۵۱	خروجی برنامه‌ی ۲.۴	۱.۴
۵۲	خروجی برنامه‌ی ۳.۴	۲.۴
۵۶	خروجی برنامه‌ی ۴.۴	۳.۴
۵۸	خروجی برنامه‌ی ۵.۴	۴.۴
۶۰	خروجی برنامه‌ی ۶.۴	۵.۴
۶۲	خروجی برنامه‌ی ۷.۴	۶.۴
۷۷	توصیف نحوه‌ی رفتار <code>switch</code>	۱.۵

۷۹	خروجی برنامه‌ی ۹.۵	۲.۵
۸۰	خروجی برنامه‌ی ۱۰.۵	۳.۵
۹۰	مدل حافظه‌ی رایانه	۱.۶
۹۳	خروجی برنامه‌ی ۱.۶	۲.۶
۹۵	خروجی برنامه‌ی ۲.۶	۳.۶
۹۶	روند ایجاد تغییرات در حافظه‌ی رایانه هنگام اجرای برنامه‌ی ۲.۶	۴.۶
۹۷	خروجی برنامه‌ی ۳.۶	۵.۶
۹۷	روند ایجاد تغییرات در حافظه‌ی رایانه هنگام اجرای برنامه‌ی ۳.۶	۶.۶
۱۰۰	خروجی برنامه‌ی ۴.۶	۷.۶
۱۰۱	تغییرات حافظه‌ی رایانه هنگام اجرای خطوط ۷ تا ۱۳ در برنامه‌ی ۴.۶	۸.۶
۱۰۲	تغییرات حافظه‌ی رایانه هنگام اجرای خطوط ۲۰ تا ۲۱ در برنامه‌ی ۴.۶	۹.۶
۱۰۴	خروجی برنامه‌ی ۵.۶	۱۰.۶
۱۰۶	تغییرات حافظه‌ی رایانه هنگام اجرای خطوط ۵ تا ۹ در برنامه‌ی ۵.۶	۱۱.۶
۱۰۷	روند تغییرات در حافظه‌ی رایانه هنگام اجرای خط ۱۳ در برنامه‌ی ۵.۶	۱۲.۶
۱۱۱	روند تخصیص حافظه به برنامه هنگام اجرای برنامه‌ی ۶.۶	۱۳.۶
۱۱۲	روند تخصیص حافظه به برنامه هنگام اجرای برنامه‌ی ۷.۶	۱۴.۶
۱۱۳	نحوه‌ی تخصیص حافظه به یک آرایه در زبان برنامه‌نویسی C/C++	۱۵.۶
۱۱۶	خروجی برنامه‌ی ۸.۶	۱۶.۶
۱۱۷	خروجی برنامه‌ی ۹.۶	۱۷.۶
۱۲۰	نحوه‌ی اختصاص حافظه به آرایه‌های دوبعدی	۱۸.۶
۱۲۲	خروجی برنامه‌ی ۱۰.۶	۱۹.۶
۱۲۴	خروجی برنامه‌ی ۱۱.۶	۲۰.۶
۱۳۲	خروجی برنامه‌ی ۱.۷	۱.۷
۱۳۵	خروجی برنامه‌ی ۲.۷	۲.۷
۱۳۶	روند تغییرات در حافظه هنگام اجرای برنامه‌ی ۱.۷	۳.۷
۱۳۸	خروجی برنامه‌ی ۴.۷	۴.۷
۱۴۱	خروجی برنامه‌ی ۵.۷	۵.۷

ع فهرست شکل‌ها

۱۴۱	۶.۷	روند تغییرات در حافظه هنگام اجرای برنامه‌ی ۵.۷
۱۴۴	۷.۷	خروجی برنامه‌ی ۶.۷
۱۴۵	۸.۷	روند تغییرات در حافظه هنگام اجرای برنامه‌ی ۶.۷
۱۴۷	۹.۷	خروجی برنامه‌ی ۷.۷
۱۵۸	۱.۸	خروجی برنامه‌ی ۴.۸
۱۶۱	۲.۸	خروجی برنامه‌ی ۵.۸
۱۶۲	۳.۸	نمایش نحوه‌ی تخصیص حافظه به متغیرهای محلی و مقداردهی آن‌ها
۱۶۳	۴.۸	نحوه‌ی تغییرات در حافظه درون یک تابع و هنگام خروج از آن
۱۶۴	۴.۸	نحوه‌ی تغییرات در حافظه درون یک تابع و هنگام خروج از آن ( ادامه)
۱۶۷	۵.۸	خروجی برنامه‌ی ۶.۸
۱۶۹	۶.۸	خروجی برنامه‌های ۷.۸ و ۸.۸
۱۷۰	۷.۸	خروجی برنامه‌های ۹.۸
۱۸۰	۱.۹	خروجی برنامه‌ی ۱.۹
۱۸۳	۲.۹	خروجی برنامه‌ی ۲.۹
۱۹۵	۱.۱۰	نشانه‌ی یک کلاس در زبان UML
۱۹۵	۲.۱۰	نمایش UML کلاس‌های شکل‌های هندسی
۲۰۶	۱.۱۱	نمایش UML کلاس دایره
۲۰۹	۲.۱۱	خروجی برنامه‌ی ۲.۱۱
۲۱۰	۳.۱۱	نحوه‌ی تخصیص حافظه هنگام ساخته‌شدن یک شی
۲۱۹	۱.۱۲	نمایش UML کلاس‌های شکل‌های هندسی
۲۲۱	۲.۱۲	نمایش UML کلاس پدر Shape برای کلاس‌های شکل‌های هندسی
۲۲۱	۳.۱۲	نمایش UML کلاس‌های شکل‌های هندسی ارث‌بری شده از کلاس Shape
۲۲۴	۴.۱۲	خروجی برنامه‌ی ۱.۱۲
۲۲۵	۵.۱۲	خروجی برنامه‌ی ۱.۱۲
۲۲۷	۶.۱۲	نمایش UML ارث‌بری متوالی

۲۳۵	۷.۱۲ خروجی برنامه‌ی ۴.۱۲
۲۳۶	۸.۱۲ خروجی برنامه‌ی ۵.۱۲
۲۴۳	۱.۱۳ نمایش یک کلاس انتزاعی (مانند Shape) در زبان UML
۲۵۵	۲.۱۳ خروجی برنامه‌ی ۳.۱۳
۲۶۰	۱.۱۴ خروجی برنامه‌ی ۱.۱۴
۲۶۳	۲.۱۴ خروجی برنامه‌ی ۲.۱۴
۲۷۴	۱.۱۵ خروجی برنامه‌ی ۴.۱۵
۲۷۶	۲.۱۵ خروجی برنامه‌ی ۵.۱۵
۲۷۹	۳.۱۵ خروجی برنامه‌ی ۶.۱۵
۲۸۴	۱.۱۶ نمایش UML کلاس Circle
۲۸۵	۲.۱۶ خروجی برنامه‌ی ۱.۱۶
۲۸۶	۳.۱۶ خروجی برنامه‌ی ۲.۱۶
۲۹۱	۴.۱۶ خروجی برنامه‌ی ۶.۱۶
۲۹۴	۵.۱۶ خروجی برنامه‌ی ۸.۱۶
۲۹۹	۱.آ مدل حافظه‌ی رایانه برای اختصاص حافظه به متغیرهای از نوع float
۳۰۳	۱.ب خروجی برنامه‌ی ۱.ب

## فهرست جدول‌ها

۱۴	کلمات کلیدی در زبان برنامه نویسی C	۱.۲
۱۴	دستورهای کامپایلر در زبان C/C++	۲.۲
۷۱	عملگرهای منطقی در زبان C/C++	۱.۵
۷۱	گزاره‌های منطقی در جبر بول	۲.۵
۱۹۴	خواص و رفتار اشیا هندسی	۱.۱۰
۲۴۸	نمونه‌ای از فایل ورودی برنامه	۱.۱۳
۲۶۹	عملگرهای ریاضی	۱.۱۵
۲۶۹	عملگرهای مقایسه	۲.۱۵
۲۷۰	عملگرهای منطقی	۳.۱۵
۲۷۰	عملگرهای تساوی ترکیبی	۴.۱۵
۲۷۰	عملگرهای پردازش بیت	۵.۱۵
۲۷۱	عملگرهای دیگر	۶.۱۵
۳۰۷	حروف آسکی غیرقابل پرینت که شامل کدهای ۰ تا ۳۱ می‌شود.	۱.پ

ق فهرست جدول‌ها

- پ.۲ حروف آسکی قابل پرینت متعارف که شامل کدهای ۳۲ تا ۱۲۷ می‌شود. . ۳۰۸
- پ.۳ حروف آسکی قابل پرینت اضافی که شامل کدهای ۱۲۸ تا ۲۵۵ می‌شود. . ۳۱۰



# فهرست برنامه‌ها

۱۲	اولین برنامه به زبان C/C++	۱.۲
۲۵	تعریف متغیرها در زبان C	۱.۳
۳۰	تأثیر بلوک بر محدوده‌ی اعتبار متغیرها	۲.۳
۳۰	ثابت‌های عددی	۳.۳
۳۱	ثابت‌های عددی	۴.۳
۳۲	برنامه‌ی مجموع چند عدد	۵.۳
۳۳	برنامه‌ی مجموع چند عدد با استفاده از آرایه‌ها	۶.۳
۳۵	ذخیره‌سازی یک رشته از حروف	۷.۳
۳۶	ذخیره‌سازی یک رشته از حروف	۸.۳
۳۷	برنامه‌ی لیست نمرات دانشجویان بدون ساختار	۹.۳
۳۸	فرمت کلی دستور struct	۱۰.۳
۳۸	تعریف یک نوع جدید با استفاده از struct	۱۱.۳
۳۸	برنامه‌ی ثبت اطلاعات دانشجو با استفاده از struct	۱۲.۳
۴۹	چاپ یک رشته ثابت روی نمایشگر (همراه با تولید خطا)	۱.۴
۵۰	چاپ یک رشته ثابت روی نمایشگر	۲.۴

۳.۴	فرمت متغیرهای مختلف برای چاپ روی نمایشگر	۵۱
۴.۴	مثال‌هایی برای نمایش گزینه‌های مختلف فرمت خروجی	۵۵
۵.۴	مثال‌هایی برای نمایش گزینه‌های مختلف ورودی اطلاعات	۵۷
۶.۴	مثال‌هایی برای نمایش گزینه‌های مختلف ورودی اطلاعات	۵۹
۷.۴	خواندن و نوشتن با استفاده از کتابخانه‌ی <code>&lt;iostream&gt;</code>	۶۱
۸.۴	خواندن و نوشتن فایل‌های متنی با استفاده از کتابخانه‌ی <code>&lt;fstream&gt;</code>	۶۳
۱.۵	فرمت دستور <code>if</code>	۶۸
۲.۵	مثالی از کاربرد دستور <code>if</code>	۶۸
۳.۵	فرمت دستور <code>if...else</code>	۶۹
۴.۵	فرمت کلی دستور <code>if</code> به طور کامل	۶۹
۵.۵	محاسبه‌ی مساحت شکل‌های هندسی با دستور <code>if</code>	۷۳
۶.۵	محاسبه‌ی مساحت شکل‌های هندسی با دستور <code>switch</code>	۷۴
۷.۵	عدم استفاده از دستور <code>break</code> در دستور <code>switch</code>	۷۵
۸.۵	فرمت کلی دستور <code>for</code>	۷۶
۹.۵	مثالی از دستور <code>for</code>	۷۸
۱۰.۵	استفاده از متغیر شمارنده‌ی حلقه‌ی <code>for</code> در اندیس آرایه‌ها	۷۹
۱۱.۵	فرمت کلی دستور <code>while</code>	۸۰
۱۲.۵	مثالی از دستور <code>while</code>	۸۱
۱.۶	استفاده از عملگر <code>&amp;</code> به منظور دسترسی به آدرس یک متغیر	۹۳
۲.۶	تعریف اشاره‌گر با استفاده از عملگر <code>*</code>	۹۴
۳.۶	استفاده از عملگر <code>*</code> برای تغییر دادن محتوای محل مورد اشاره	۹۵
۴.۶	استفاده از اشاره‌گر به عنوان یک متغیر	۹۹
۵.۶	استفاده از کلمه‌ی کلیدی <code>new</code> جهت اختصاص حافظه به یک اشاره‌گر	۱۰۳
۶.۶	بررسی نشستی حافظه در برنامه‌نویسی	۱۰۸
۷.۶	بررسی نشستی حافظه در برنامه‌نویسی	۱۱۰
۸.۶	اشاره کردن به تک‌تک عناصر یک آرایه	۱۱۴
۹.۶	اشاره کردن به کل عناصر یک آرایه	۱۱۶

۱۲۰	اشاره کردن به یک آرایه‌ی دوبعدی	۱۰.۶
۱۲۳	struct به یک	۱۱.۶
۱۳۱	استفاده از کلمه‌ی کلیدی new جهت تولید یک آرایه‌ی دینامیکی	۱.۷
۱۳۳	نحوه‌ی آزاد کردن حافظه‌ی اختصاص یافته به یک آرایه‌ی دینامیکی	۲.۷
۱۳۴	روش نادرست افزودن به تعداد عناصر یک آرایه‌ی دینامیکی	۳.۷
۱۳۷	روش صحیح افزودن به تعداد عناصر یک آرایه‌ی دینامیکی	۴.۷
۱۴۰	روش تولید آرایه‌های دوبعدی پراکنده در حافظه	۵.۷
۱۴۳	روش تولید آرایه‌های دوبعدی یکپارچه در حافظه	۶.۷
۱۴۵	روش تولید آرایه‌های چندبعدی پراکنده در حافظه	۷.۷
۱۵۲	تعریف تابع در زبان C	۱.۸
۱۵۳	استفاده از کلمه‌ی کلیدی void	۲.۸
۱۵۴	استفاده از کلمه‌ی کلیدی void	۳.۸
۱۵۶	کاربرد اعلام و تعریف در زبان C/C++	۴.۸
۱۵۹	روش‌های مختلف ارسال آرگومان به یک تابع	۵.۸
۱۶۵	تفاوت نوشتاری بین روش‌های مختلف ارسال آرگومان به تابع	۶.۸
۱۶۷	مثالی برای ارائه‌ی مفهوم چندریختی	۷.۸
۱۶۸	بازنویسی برنامه‌ی ۷.۸ با استفاده از مفهوم توابع چندریختی	۸.۸
۱۶۹	مثالی برای استفاده از توابع چندریختی	۹.۸
۱۸۰	کار با رشته‌های متنی با استفاده از کتابخانه‌ی <string>	۱.۹
۱۸۲	کار با رشته‌های متنی با استفاده از کتابخانه‌ی <string>	۲.۹
۲۰۴	نحوه‌ی تعریف کلاس در زبان C++	۱.۱۱
۲۰۶	تعریف کلاس Circle مطابق شکل ۱.۱۱	۲.۱۱
۲۰۸	عدم حساسیت تقدم و تأخر تعریف در صفت‌ها و رفتارهای درونی کلاس	۳.۱۱
۲۱۱	نحوه‌ی استفاده از اشاره‌گر به یک شی	۴.۱۱
۲۲۲	نحوه‌ی اعمال وراثت در زبان C++	۱.۱۲

۲۲۷	مثالی از ارث‌بری متوالی
۲۳۱	استفاده از اشاره‌گر در ارث‌بری
۲۳۴	بازنویسی رفتارهای ارث‌بری شده با استفاده از کلمه‌ی کلیدی virtual
۲۳۵	بازنویسی رفتارهای ارث‌بری شده با استفاده از کلمه‌ی کلیدی virtual
۲۴۳	تعریف یک کلاس انتزاعی
۲۴۵	بازنویسی رفتارهای انتزاعی
۲۵۰	متن کامل برنامه‌ی شکل‌های هندسی
۲۵۸	سطوح مختلف حفاظت و دسترسی به داده‌ها
۲۶۰	سطح دسترسی خصوصی
۲۶۲	سطوح مختلف حفاظت و دسترسی به داده‌ها
۲۶۴	سطوح مختلف حفاظت و دسترسی به داده‌ها
۲۶۷	مثالی برای نمایش عملگرهای یک‌طرفی
۲۶۸	مثالی برای نمایش عملگرهای دوطرفی
۲۷۱	مثالی برای نشان دادن نیاز به سربارگذاری عملگرها
۲۷۳	سربارگذاری عملگرهای یک‌طرفی
۲۷۵	سربارگذاری عملگرهای دوطرفی
۲۷۷	استفاده از چندریختی در سربارگذاری عملگرها
۲۸۴	استفاده از رفتارهای یک کلاس بدون مقدار اولیه دادن به صفت‌های آن
۲۸۵	استفاده از رفتار Initialize برای مقدار اولیه دادن به خواص یک کلاس
۲۸۷	فرمت کلی یک سازنده
۲۸۷	استفاده از سازنده برای کلاس Circle
۲۸۹	مقایسه‌ی زمان‌هایی که یک سازنده فراخوانی می‌شود
۲۹۰	تعریف سازنده‌های غیر پیش‌فرض
۲۹۲	فرمت کلی سازنده و مخرب در زبان C++
۲۹۳	استفاده از مخرب قبل از حذف شی از حافظه

ن فهرست برنامه‌ها

ب.۱ ارسال تابع به عنوان آرگومان به یک تابع دیگر . . . . . ۳۰۲