





دانشگاه پیام نور استان تهران

مرکز / واحد پرند

گروه فنی و مهندسی

پروژه ی کارشناسی

رشته ی مهندسی کامپیوتر

گرایش نرم افزار

عنوان پروژه:

دفترچه تلفن به زبان C++

استاد راهنما:

سید علی رضوی ابراهیمی

تهیه کننده:

فاطمه اکبرشاهی

مهر ماه ۱۴۰۰

کلیه حقوق مادی مترتب بر نتایج مطالعات،
ابتکارات و نوآوری های ناشی از این پروژه
متعلق به :

"دانشگاه پیام نور استان تهران /مرکز پرند"

می باشد.

چکیده

همان طور که همه می دانید ، در دنیای دیجیتال امروز انجام بسیاری از کارها بدون کامپیوتر غیرممکن است ، زیرا کاربرد کامپیوتر در زندگی بسیار زیاد است که آن را به بخشی جدایی ناپذیر از زندگی تبدیل می کند. تقریباً همه با کامپیوتر و اینترنت آشنایی دارند. استفاده از کامپیوتر عملاً توسط کلیه موسساتی که در همه ادارات ، مدارس ، کالج ها و غیره استفاده می کنند انجام می شود زیرا از ویژگی همه کاره بودن برخوردار است. با توسعه و فن آوری به روز شده ، کامپیوتر تقریباً در همه زمینه ها کاربرد داشته و هر شخصی در خانه خود استفاده می کند. پس نتیجه میگیریم هر چه انسان بیشتر بتواند با آن ارتباط برقرار کند به روز تر است.

حال چگونه میتوان بیشتر با آن مانوس شد یا زبان آن را فرا گرفت ؟

کد و برنامه نویسی، زبان کامپیوتر است؛ که با آن میشود با کامپیوتر صحبت کرد و زبانش را فراگرفت. برنامه نویسی به شما یاد می دهد که چگونه فکر کنید. اهمیت برنامه نویسی در دنیای امروز که همه چیز وابسته به تکنولوژی و کامپیوتر است، بالا رفته. دانستن علم برنامه نویسی به شما دیدگاه یک مهندس نرم افزار را می دهد و به شما می آموزد که چگونه دید کامپیوتری داشته باشید و مشکلات را از آن منظر حل کنید. زبان برنامه نویسی بر پایه ی منطق، الگوریتم و ریاضیات است و به شما آموزش می دهد که برای حل مشکلات بزرگ، لازم است که آن ها را به بخش های کوچکتر تقسیم کنید تا آسان تر حل شوند .

در عصر حاضر که تکنولوژی در حال بلعیدن همه چیز است، درک بهتر از آن مزیت بزرگی محسوب می شود. دانستن تکنولوژی (به طور مثال استفاده ی کاربردی از تلفن های هوشمند و برنامه ها) سبب عوض شدن نگرش انسان به این دنیا می شود. کدنویسی نه تنها به شما طریقه ی ساخت یک برنامه را شرح می دهد، بلکه شما را از سرعت بالای توسعه تکنولوژی آگاه می سازد و به شما می آموزد چرا یک برنامه ساخته شده است؟ چرا سازندگان، این برنامه را ساخته اند و چرا دیگران این موقعیت را برای ساخت این برنامه درک نکرده اند؟

یادگیری یک زبان برنامه نویسی فرقی نمی کند چه زبانی باشد، (HTML، C++، ...) برقراری ارتباط شما را با توسعه دهنده ها آسان تر می کند. همچنین یادگیری برنامه نویسی به درک و حل الگوریتم های پیچیده، ارتباط و انتقال مفاهیم پایه و اساسی کمک شایانی می کند.

فهرست مطالب

صفحه	عنوان
ج	چکیده
ذ	پیشگفتار
۱	فصل اول
۲	۱-۱ ایجاد زبان برنامه نویسی C++
۲	۱-۱-۱ نام C++
۳	۱-۱-۲ فلسفه ایجاد C++
۴	۱-۲-۱ توابع C++
۴	۱-۲-۱-۱ تابع کتابخانه
۵	۱-۲-۲-۱ تابع تعریف شده از سوی کاربر
۶	۱-۲-۳-۱ اعلان پروتوتایپ تابع
۸	۱-۲-۴-۱ فراخوانی تابع
۸	۱-۲-۵-۱ تعریف تابع
۹	۱-۲-۶-۱ ارسال آرگومان به تابع
۹	۱-۲-۷-۱ گزاره بازگشت

۱۰	۳-۱-انواع متغیر ها در C++
۱۰	۴-۱-انواع عملگرها در C++
۱۰	۱-۴-۱-عملگرهای محاسباتی
۱۱	۲-۴-۱-عملگرهای مقایسه ای
۱۱	۳-۴-۱-عملگر های منطقی
۱۱	۴-۴-۱-عملگرهای بی تی
۱۲	۵-۴-۱-عملگر های متفرقه
۱۳	۵-۱-آموزش نصب محیط برنامه نویسی C++ (dev c++)
۱۷	۶-۱-جمع بندی فصل اول
۱۸	فصل دوم
۱۹	۱-۲-معرفی کتابخانه های برنامه دفترچه تلفن
۲۰	۲-۲-نوشتن تابع main برنامه
۲۱	۱-۲-۲-حلقه while داخل تابع main
۲۲	۱-۲-۲-۱-دستور switch case
۲۳	۲-۲-۱-۲-تعریف Error در برنامه
۲۴	۳-۲-تعریف struct (ساختار)
۲۵	۴-۲-نوشتن تابع برای ایجاد فایل
۲۹	۵-۲-نوشتن تابع برای گرفتن اطلاعات (نام و شماره تلفن) از کاربر

۳۱	۶-۲- نوشتن تابع برای لیست کردن اطلاعات کاربر
۳۴	۷-۲- نوشتن تابع برای جستجو کردن اطلاعات
۳۶	۸-۲- نمایش تمام کد های دفترچه تلفن به صورت یکجا
۴۰	۹-۲- جمع بندی فصل دوم
۴۱	فصل سوم
۴۲	۱-۳- نمایش نحوه اجرای برنامه
۴۷	۲-۳- کاربرد اصول مهندسی نرم افزار در طراحی دفترچه تلفن
۴۸	۳-۳- کاربرد فرایندهای مهندسی نرم افزار در طراحی پروژه
۵۰	۴-۳- چگونه دانش مهندسی نرم افزار به طراحی پروژه کمک نمود ؟
۵۱	۵-۳- جمع بندی فصل سوم
۵۲	جمع بندی و ارائه پیشنهادات

پیشگفتار

این مقاله در ارتباط با زبان برنامه نویسی C++ نگارش شده است. ما در این مقاله که در طی چند فصل بخش بندی شده سعی بر داشتیم که خواننده با این زبان برنامه نویسی آشنا شده و همچنین آموزش نوشتن برنامه ای با این زبان، به منظور یادگیری بهتر و کاربردی تر این زبان برنامه نویسی برای شما عزیزان شرح داده شده است. سی پلاس پلاس به عنوان قدرتمندترین زبان برنامه نویسی تا به کنون است که به جرأت می توان گفت به عنوان یک زبان برنامه نویسی غالب بر دیگر زبان های برنامه نویسی لقب «هیولای زبان های برنامه نویسی» را به خود اختصاص می دهد. با توجه به ساختار و نقشه راه توسعه خود، هنوز هم به عنوان یکی از پر طرفدارترین و پر کاربردترین زبان های برنامه نویسی ساخت دست بشر به شمار می رود. پیاده سازی برنامه ای که به شما آموزش داده میشود برای کسانی که ابتدای راه یادگیری این زبان برنامه نویسی گسترده هستند بسیار کاربردی است.

فصل اول

مقدمه

در این فصل با تاریخچه زبان C++ آشنا خواهید شد و همچنین انواع توابع در این زبان برنامه نویسی برای شما شرح داده خواهد شد. در ادامه مطالب نیز انواع عملگرها و متغیرهای موجود در C++ به شما معرفی می شود. در پایان مطالب این فصل نیز نصب یکی از پر استفاده ترین محیط های برنامه نویسی C++ به شما آموزش داده می شود.

۱-۱ ایجاد زبان برنامه نویسی C++

استراس تروپ کار بر روی زبان « C دارای کلاس^۱ » را در سال ۱۹۷۹ آغاز کرد. ایده ی ساخت این زبان جدید در زمان کار بر روی تز دکترای خود به ذهن استراس تروپ خطور نمود. او متوجه شد که سیمولا^۲ دارای ویژگی هایی مناسب برای ساخت برنامه های بسیار بزرگ است اما برای استفاده ی عملی بسیار کند است اما بی سی پی ال^۳ با وجود سرعت بسیار زیاد برای ساخت برنامه های بزرگ بسیار سطح پایین است. زمانی که استراس تروپ کار خود را در آزمایشگاه های بل^۴ آغاز نمود با مشکل تحلیل هسته یونیکس با توجه به محاسبات توزیع شده روبرو شده بود. با یادآوری تجربیات خود در دوران دکترا، او زبان C را با استفاده از ویژگی های سیمولا گسترش داد .

C به این دلیل انتخاب شد که یک زبان عمومی، سریع، قابل حمل، و در سطح گسترده ای در حال استفاده بود. در ابتدا ویژگی های کلاس، کلاس های مشتق شده، کنترل نوع قوی، توابع درون خطی و آرگومان های پیش فرض از طریق C front به C اضافه شد. در سال ۱۹۸۳ نام زبان از « C با کلاس » به C++ تغییر یافت. ویژگی های دیگر شامل توابع مجازی، سربار گذاری عملگر و نام تابع، ارجاعات، ثوابت، کنترل حافظه توسط کاربر به صورت آزاد، کنترل نوع بهتر، و توضیحات یک خطی به صورت BCPL با استفاده از «//» نیز به آن اضافه شد. در سال ۱۹۸۵ اولین نسخه زبان برنامه نویسی C++ انتشار یافت و مرجع مهمی برای این زبان فراهم شد

۱-۱-۱ نام C++

C++ توسط استراس تروپ ریاضی دان دانمارکی در سال ۱۹۷۹ برای بهبود زبان سی و بر مبنای آن ساخته شد و آن را « C با کلاس » نام گذاری نمود. در سال ۱۹۸۳ به C++ تغییر نام داد . در واقع این نام منسوب به ریک ماسکیتی (اواسط ۱۹۸۳) است و برای اولین بار در دسامبر سال ۱۹۸۳ به کار برده شد.

۱- C with classes

۲-simula

۳-Basic Combined Programming Language

۴-Bell Labs

وقتی که به صورت خصوصی از ریک ماسکیتی در مورد این اسم سؤال شد او در جواب گفت که این اسم به صورت خودمانی در بین آن‌ها به کار می‌رفته است و تصور نمی‌کردند که این نام به صورت نام رسمی این زبان درآید. در طول مدت تحقیق این زبان بنام «C جدید» و بعدها «C با کلاس» خوانده شد. در علوم کامپیوتر هنوز هم ++C به عنوان ابرساختار C شناخته می‌شود. آخرین نام از عملگر ++ در زبان C (که برای افزایش مقدار متغیر به اندازه‌ی یک واحد بکار می‌رود) و یک عرف معمول برای نشان دادن افزایش قابلیت‌ها توسط + ناشی گشته است. با توجه به نقل قولی از استراس تروپ: «این نام ویژگی‌ها تکاملی زبان در C را نشان می‌دهد.» ++C نام زبانی غیر مرتبط به این زبان است.

۱-۱-۲- فلسفه ایجاد ++C

- ++C طراحی شده است تا یک زبان عمومی با کنترل نوع ایستا و همانند C قابل حمل و پربازده باشد.
- ++C طراحی شده است تا مستقیماً و به صورت جامع از چندین شیوه برنامه‌نویسی بتوان از آن استفاده کرد (برنامه‌نویسی ساخت یافته، برنامه‌نویسی شی گرا، انتزاع داده، و برنامه‌نویسی جنریک).
- ++C طراحی شده است تا به برنامه‌نویس امکان انتخاب دهد حتی اگر این انتخاب اشتباه باشد.
- ++C طراحی شده است تا حداکثر تطابق با C وجود داشته باشد و یک انتقال راحت از C را ممکن سازد.
- ++C از بکاربردن ویژگی‌های خاص که مانع از عمومی شدن است خودداری می‌نماید.
- ++C از ویژگی‌هایی که بکار برده نمی‌شوند استفاده نمی‌کند.
- ++C طراحی شده است تا بدون یک محیط پیچیده عمل نماید.

۱-۲-توابع C++

در برنامه نویسی تابع به یک سری کد خاص اشاره می کند که معمولاً در بالای سایر کدها نوشته می شود و در برنامه با صدا زدن اسم تابع به آن رجوع می شود . منظور از تابع گزاره ای است که کدها را برای اجرای وظیفه خاصی گروه بندی می کند . در این بخش در ارتباط با اینکه چه نوع تابع هایی در زبان برنامه نویسی C++ وجود دارند مطالبی را ارائه میکنیم .

بسته به این که تابع از قبل تعریف شده باشد یا از سوی برنامه نویس ایجاد شود دو نوع تابع وجود دارد:

- تابع کتابخانه
- تابع تعریف شده از سوی کاربر

۱-۲-۱-تابع کتابخانه^۱

تابع های کتابخانه تابع های داخلی زبان برنامه نویسی C++ هستند. برنامه نویس می تواند با فراخوانی مستقیم تابع از این تابع های کتابخانه استفاده کند و نیازی نیست که آن ها را خودش بنویسد.

```
#include <iostream>

#include <cmath>

using namespace std;

int main()
{
    double number, squareRoot;

    cout << "Enter a number: ";

    cin >> number;
```

مثال ۱:

```
// sqrt() is a library function to calculate square root

squareRoot = sqrt(number);

cout << "Square root of " << number << " = " << squareRoot;

return 0;

{
```

Enter a number: 26

Square root of 26 = 5.09902

خروجی:

تابع کتابخانه `sqrt()` برای محاسبه ریشه مربع یک عدد استفاده می‌شود. به کد `#include <cmath>` در برنامه فوق توجه کنید. در این کد `cmath` یک فایل «هدر» (Header) است. تعریف تابع `sqrt()` یعنی بدنه تابع در فایل هدر `cmath` قرار دارد. زمانی که فایل `cmath` را با استفاده از دستور `#include <cmath>` در برنامه خود بگنجانید، می‌توانید از همه تابع‌های تعریف شده در آن استفاده کنید.

۱-۲-۲- تابع تعریف شده از سوی کاربر

C++ به برنامه‌نویس امکان می‌دهد که تابع خاص خود را تعریف کند. تابع تعریف شده از سوی کاربر به گروه‌بندی کد می‌پردازد تا وظیفه خاصی را اجرا کند و این گروه کد یک نام (شناسه) دارد. زمانی که تابع از هر بخش از برنامه فراخوانی شود، کدهای تعریف شده در بدنه تابع را اجرا می‌کند.

```
void function_name() {
    ... ..
    ... ..
}

int main() {
    ... ..
    function_name();
    ... ..
}
```

شکل ۱-۱ طرز کار تابع‌های تعریف شده از سوی کاربر

زمانی که یک برنامه شروع به اجرا می‌کند، سیستم تابع `main()` را فراخوانی می‌کند، یعنی سیستم شروع به اجرای کد از تابع `main()` می‌کند. زمانی که کنترل برنامه به `function_name()` درون `main()` برسد، به `void function_name()` انتقال می‌یابد و همه کدهای درون آن را اجرا می‌کند. سپس کنترل برنامه مجدداً به تابع `main` بازمی‌گردد و چنان که در تصویر فوق مشخص است، کد پس از فراخوانی `function_name()` اجرا می‌شود.

۱-۲-۳-اعلان پروتوتایپ تابع

اگر یک تابع تعریف شده از سوی کاربر پس از تابع `main()` تعریف شود، کامپایلر خطایی نمایش خواهد داد. دلیل این امر آن است که کامپایلر از وجود تابع تعریف شده از سوی کاربر، نوع آرگومان‌های ارسالی به تابع و نوع بازگشتی آن ناآگاه است. پروتوتایپ تابع در `C++` یک اعلان از تابع بدون بدنه است که اطلاعاتی را در مورد تابع تعریف شده از سوی کاربر در اختیار کامپایلر قرار می‌دهد.

مثال ۲ (در برنامه دو عدد صحیح با تابع `add()` جمع میشوند و در تابع `main()` نمایش داده میشوند):

```
#include <iostream>

using namespace std;

// Function prototype (declaration)

int add(int, int);

int main()

{
```

```

int num1, num2, sum;

cout<<"Enters two numbers to add: ";

cin >> num1 >> num2;

// Function call

sum = add(num1, num2);

cout << "Sum = " << sum;

return 0;

}

// Function definition

int add(int a, int b)

{

    int add;

    add = a + b;

    // Return statement

    return add;

}

```

Enters two integers: 8

-4

Sum = 4

خروجی:

پروتوتایپ تابع در مثال فوق به صورت زیر است:

```
int add (int ،int);
```


همان طور که مشخص است در پروتوتایپ تابع هیچ بدنه‌ای ندارد. ضمناً تنها نوع بازگشتی آرگومان‌ها ذکر شده و خبری از خود آرگومان‌ها نیست. پروتوتایپ تابع^۱ را می‌توان به شکل زیر نیز اعلان کرد، اما لزومی به نوشتن آرگومان‌ها وجود ندارد:

```
int add(int a, int b);
```

۱-۲-۴- فراخوانی تابع

برای اجرای کد بدنه تابع، تابع تعریف شده از سوی کاربر باید احضار یا فراخوانی شود. در مثال ۲ دستور زیر درون main() تابع تعریف شده کاربر را فراخوانی می‌کند. بدین ترتیب تابع یک عدد صحیح بازگشت می‌دهد که در متغیر add ذخیره می‌شود.

```
add(num1, num2);
```

۱-۲-۵- تعریف تابع

خود تابع به صورت تعریف تابع مورد ارجاع قرار می‌گیرد. زمانی که تابعی فراخوانی می‌شود، کنترل به گزاره اول بدنه تابع انتقال می‌یابد. سپس گزاره‌های دیگر در بدنه تابع به ترتیب اجرا می‌شوند. هنگامی که همه کدهای درون تعریف تابع اجرا شدند، کنترل برنامه به برنامه فراخوانی کننده بازگشت می‌یابد. تعریف تابع در مثال ۲ به صورت زیر است:

```
// Function definition
int add(int a,int b)
{
    int add;
    add = a + b;
    return add;
}
```

۱-prototype (شرحی از نام، نوع برگشتی و نوع آرگومانهای تابع به کامپایلر ارائه می‌دهد).

۱-۲-۶-ارسال آرگومان به تابع

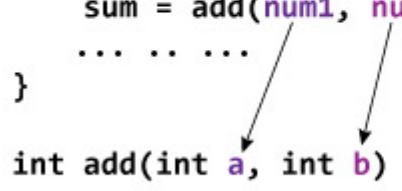
پارامتر آرگومان در برنامه نویسی، اشاره به داده هایی دارد که در زمان فراخوانی تابع به آن ارسال می شوند. در مثال فوق، دو متغیر `num1` و `num2` در طی فراخوانی تابع به آن ارسال می شوند. این آرگومان ها به نام آرگومان های واقعی شناخته می شوند. مقدار `num1` و `num2` به ترتیب با متغیرهای `a` و `b` مقداردهی می شوند. این آرگومان های `a` و `b` به نام آرگومان های صوری شناخته می شوند.

```
# include <iostream>
using namespace std;

int add(int, int);

int main() {
    ... ..
    sum = add(num1, num2); // Actual parameters: num1 and num2
    ... ..
}

int add(int a, int b) { // Formal parameters: a and b
    ... ..
    add = a+b;
    ... ..
}
```

A diagram with two arrows. One arrow starts from the 'num1' parameter in the function call 'sum = add(num1, num2);' and points down to the 'a' parameter in the function definition 'int add(int a, int b)'. The other arrow starts from the 'num2' parameter in the function call and points down to the 'b' parameter in the function definition.

شکل ۱-۲ نمایش ارسال آرگومان به تابع

۱-۲-۷-گزاره بازگشت

هر تابعی می تواند با استفاده از گزاره `Return` یک مقدار منفرد به برنامه فراخوانی کننده بازگشت دهد. در مثال ۲، مقدار `add` از تابع تعریف شده از سوی کاربر با استفاده از `return add;` به برنامه فراخوانی کننده بازگشت می یابد. سپس این مقدار در متغیر `sum` ذخیره می شود. توجه داشته باشید که متغیر بازگشت یافته یعنی `add` از نوع `int` و متغیر `sum` نیز از نوع `int` است.

۱-۳- انواع متغیر ها در C++

متغیر نوع int: برای تعریف متغیر از جنس صحیح از int استفاده می کنیم. محدوده متغیر از جنس صحیح (int) از ۳۲۷۰۸- تا ۳۲۷۰۷+ است.

متغیر نوع float: ما اگر بخواهیم از اعداد اعشاری استفاده کنیم متغیر نوع float را به کار می گیریم.

متغیر نوع double: این متغیر مانند نوع float عمل می کند با این تفاوت که متغیر از نوع double محدوده بیشتری را به ما می دهد و برای مقادیر بزرگ تر استفاده می شود. سائز متغیر نوع float چهاربایت^۱ و متغیر نوع double هشت بایت است.

متغیر نوع char^۲: اگر ما بخواهیم از متغیری استفاده کنیم که کاراکترها را به ما نشان دهد برای این کار ما باید از متغیر نوع char استفاده کنیم که می تواند یک حرفی و چند حرفی باشد. متغیر از نوع char یک بایت از فضای سیستم را اشغال می کند.

نکته: اگر ما بخواهیم محدوده ما فقط اعداد نا منفی باشد از unsigned استفاده می کنیم.

مثل: unsigned int

متغیر نوع bool: متغیر نوع bool دارای دو مقدار true و false است.

۱-۴- انواع عملگرها در C++

۱-۴-۱- عملگرهای محاسباتی

۱-بایت (Byte): یکی از یکاهای اساسی سنجش مقدار داده ها در رایانه و به معنی هشت بیت متوالی است.

۲-character:char: کاراکتر

مثال	مفهوم	علامت معادل در C++	علامت در ریاضیات
$x+y$	جمع	+	+
$x-y$	تفریق	-	-
$x*y$	ضرب	*	\times یا \cdot
x/y	تقسیم	/	\div
$x\%y$	باقیمانده	%	mod

جدول ۱-۱

۱-۴-۲- عملگرهای مقایسه ای

مثال	مفهوم	علامت معادل در C++	علامت در ریاضیات
$x>y$	بزرگتر از	>	>
$x<y$	کوچکتر از	<	<
$x>=y$	بزرگتر مساوی	>=	\geq
$x<=y$	کوچکتر مساوی	<=	\leq
$x==y$	مساوی	==	=
$x!=y$	نا مساوی	!=	\neq

جدول ۲-۱

۱-۴-۳- عملگرهای منطقی

مثال	مفهوم	علامت معادل در C++	علامت در ریاضیات
$x>0 \ \&\& \ x<10$	ترکیب عطفی (و)	&&	\vee
$x>0 \ \ y>0$	ترکیب فصلی (یا)		\wedge
$!(x>0)$	نقیض	!	!

جدول ۳-۱

۱-۴-۴- عملگرهای بی تی

علامت در C++	مفهوم
&(and BT)	حاصل & دو چیز وقتی یک است که هر دو یک باشند.
(or BT)	حاصل دو چیز وقتی یک است که حداقل یکی از آن دو چیز یک باشد.
^(not BT)	این عملگر حاصل را برعکس می کند (صحیح به غلط و غلط به صحیح تبدیل می شود).
<<	(شیفت به چپ) مقدار عبارت را دو برابر می کند.
>>	(شیفت به راست) مقدار عبارت را نصف می کند.
^	(یا انحصاری) اگر هر دو عبارت با هم برابر نباشند حاصل ۱ است (حاصل ۰، ۱، ۰، ۱ یک است)

جدول ۱-۴

۱-۴-۵- عملگرهای متفرقه

عملگرهای ترکیبی: این نوع عملگرها حاصل ترکیب بعضی علامت ها با = است و مزیت این روش این است که ما نام متغیر را یک بار می نویسیم و اگر نام متغیر ما طولانی باشد این نوع عملگرها به ما کمک می کنند همچنین استفاده کردن از این عملگرها خوانایی برنامه توسط خودمان را بالا می برد مثال: عملگرهای ترکیبی شامل +=، /=، *=، و -= هستند.

عملگر؟ (عملگر شرطی): عملگر؟ عملگری است که مقایسه را انجام می دهد و شبیه کار دستورات if و else است.

این عملگر برای این استفاده می رود که به جای این که چند دستور را مجزا بنویسیم می توانیم آن ها را باهم و در یک دستور بنویسیم مثال:

```
//x=10; y=x+4;
```

```
y=(x=10, x+4)
```

۱-۵- آموزش نصب محیط برنامه نویسی C++ (dev c++)

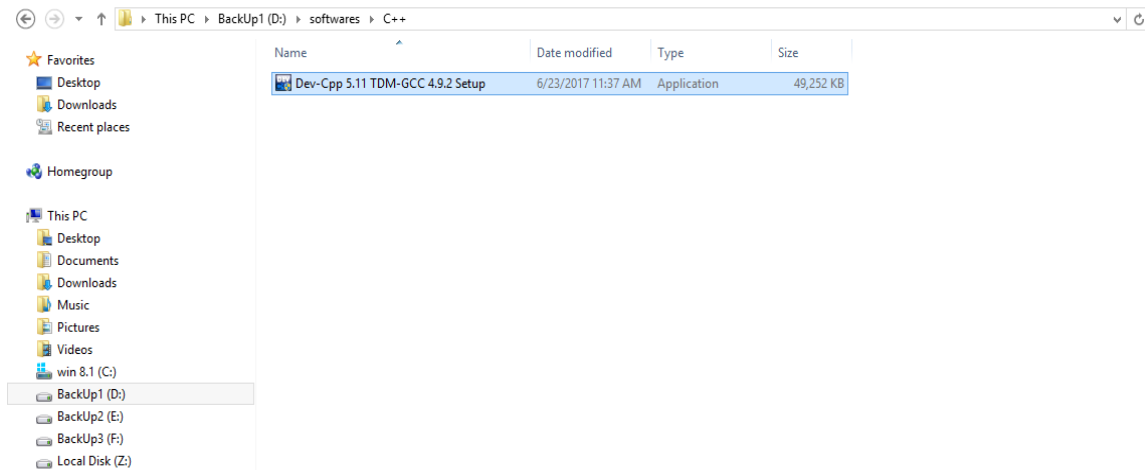
برای نوشتن و کامپایل کردن کد به زبان C++ محیط های متفاوتی در سیستم عامل های متفاوت وجود دارد از جمله : Turbo C++ ، Borland C++ ، Microsoft C++ ، Code Blocks ، kdevelop ، Anjuta ، Code Lite ، Qt و ...

اما در بین تمامی این محیط ها برای انجام پروژه محیط dev C++ را انتخاب کردیم به دلیل اینکه یک ادیتور رایگان می باشد ، هم به زبان C و هم به زبان C++ می توان در آن برنامه نویسی کرد ، محیطی کاملاً سبک و قابل پسند از لحاظ گرافیکی می باشد و برنامه نویسان زیادی در سر تا سر جهان از این ادیتور استفاده می کنند .

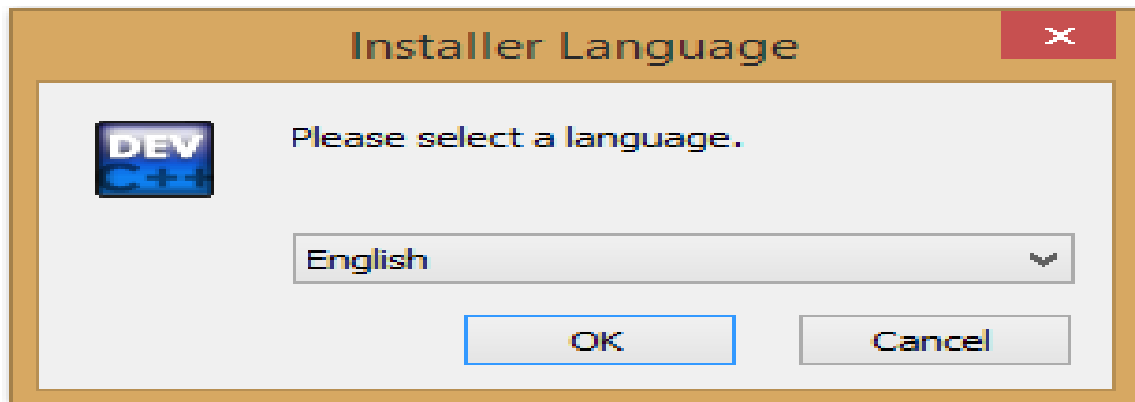
آموزش نصب dev c++ :

۱- ابتدا نرم افزار را از سایت دانلود کنید .

۲- پس از اتمام دانلود بر روی آیکن مربوط به نرم افزار ++C کلیک کرده تا مراحل نصب آغاز شود .



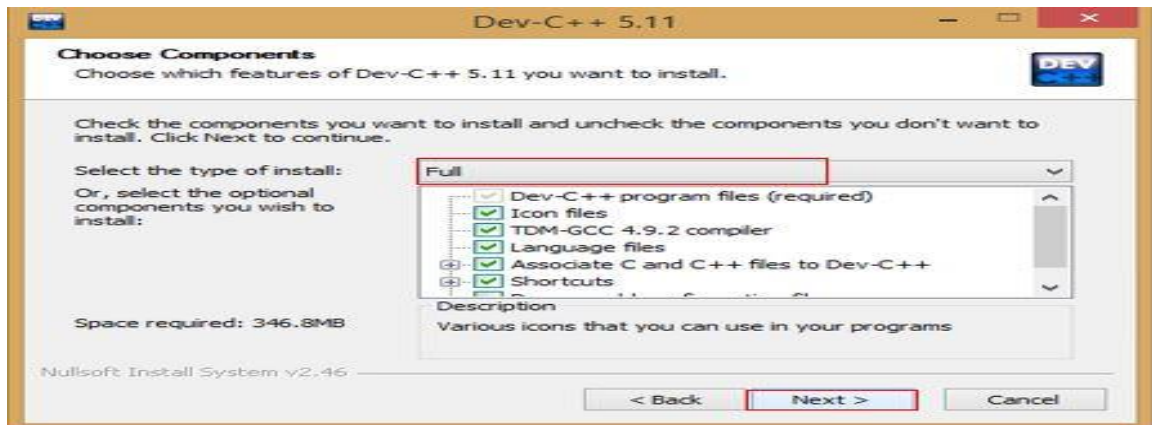
۳- در ادامه ی کار صفحه ای ظاهر می شود که باید زبان مورد نظر را انتخاب نماییم . زبان انگلیسی به طور پیش فرض انتخاب شده است پس بدون انجام تغییر بر روی Ok کلیک می کنیم.



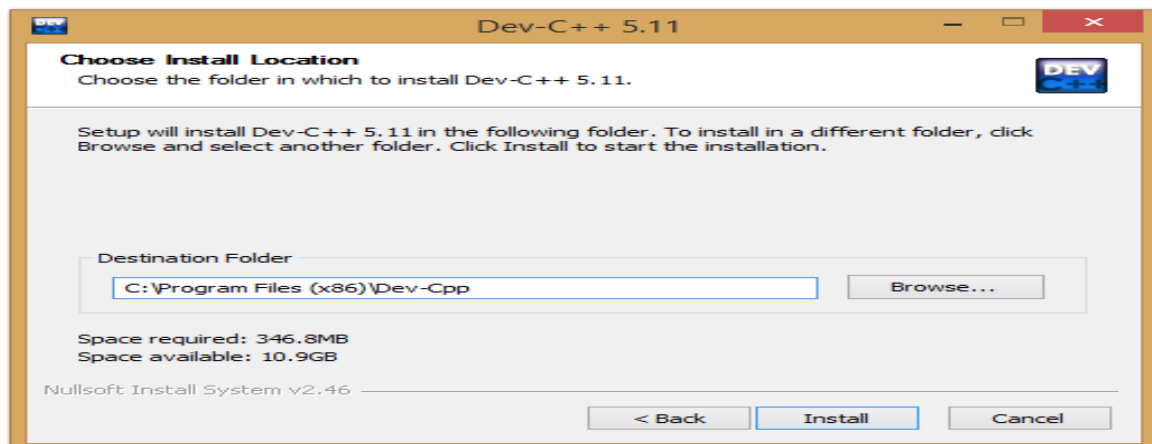
۴- سپس صفحه ای ظاهر می شود که باید روی گزینه I Agree کلیک نمایید .



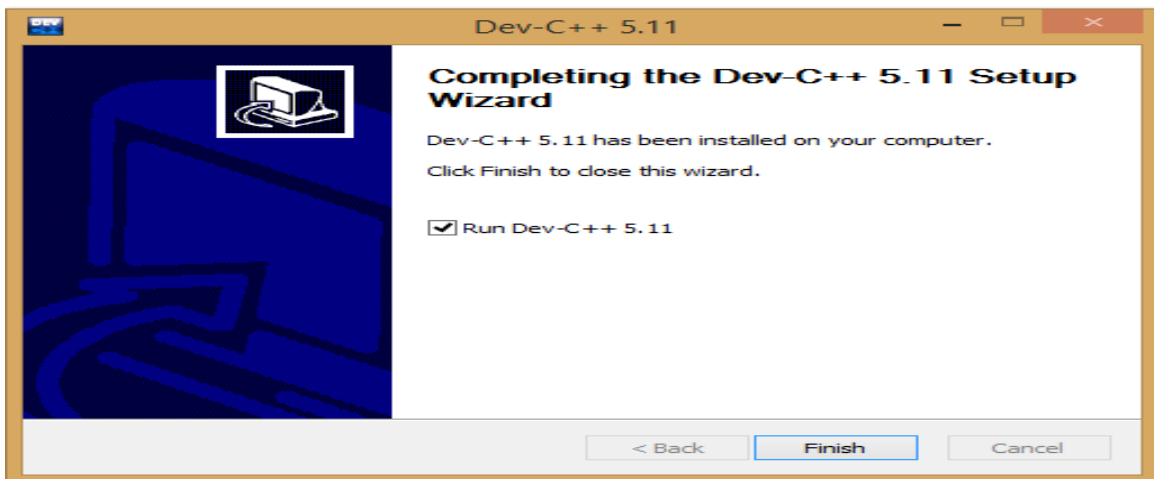
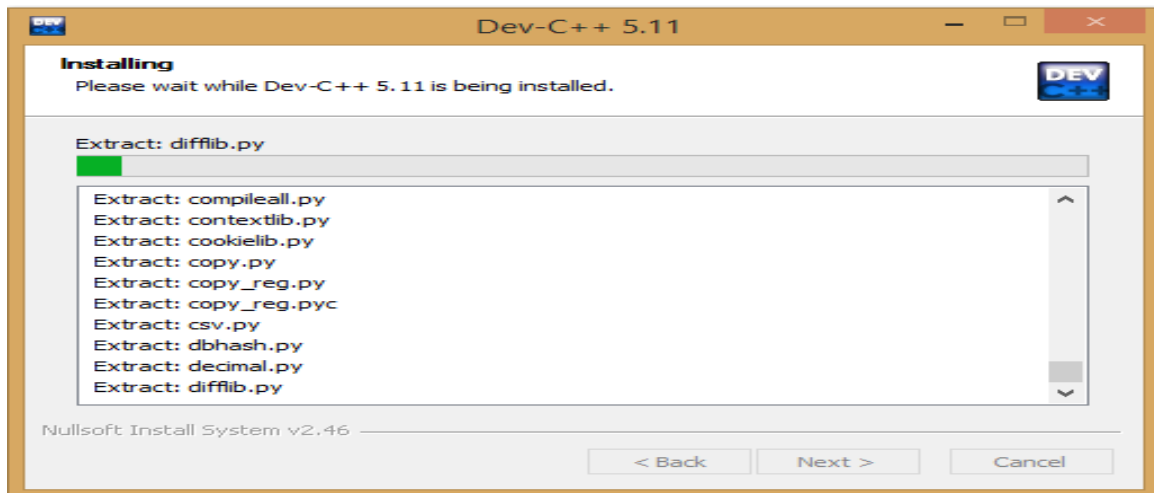
۵- در قسمت بعدی در پنجره ظاهر شده می توان آیکن ها و قسمت هایی از نرم افزار C++ را که نصب می شود را به دلخواه انتخاب نمود. به منظور نصب کامل بر روی select the type of install full کلیک کرده و بر روی next کلیک نمایید .



۶- در پنجره بعد باید محل نصب نرم افزار C++ را انتخاب کنیم . (حداقل فضای مورد نیاز جهت نصب، ۳۴۷ مگابایت می باشد.) پس از انجام این کار بر روی گزینه Install کلیک کرده تا نصب نرم افزار سی پلاس پلاس آغاز گردد .



۷- پس از کلیک بر روی گزینه Install منتظر می مانیم تا نصب برنامه C++ به پایان برسد. سپس بر روی Finish کلیک می کنیم .



۱-۶- جمع بندی فصل اول

در این فصل در ابتدا یاد گرفتیم که زبان برنامه نویسی C++ چگونه ایجاد شد ، چگونه نام گذاری شد و فلسفه ایجاد آن چه بود . سپس انواع توابع C++ به شما معرفی شد که این توابع به طور کلی به دو دسته ی توابع کتابخانه ای و توابع تعریف شده توسط کاربر دسته بندی می شدند و توابع تعریف شده توسط کاربر خود شامل قسمت های مختلفی بود که برای شما عزیزان شرح داده شد . در این فصل انواع متغیر ها (int, double, float, ...) در زبان سی پلاس پلاس به شما معرفی شد . انواع عملگر ها (منطقی ، مقایسه ای ، محاسباتی ، بی تی و ...) در این زبان نیز برای درک بهتر در قالب جدول به شما معرفی شد . در پایان کار نیز نصب dev c++ (محیطی که پروژه ما در آن نوشته شده) که محیطی کاملاً سبک از نظر گرافیکی برای برنامه نویسی C++ می باشد به شما آموزش داده شد .

فصل دوم

مقدمه

در این فصل قصد داریم پروژه‌ی دفترچه تلفنی را که طراحی کردیم به صورت کاملاً جزئی و دقیق برای شما شرح دهیم. تمامی کدهای نوشته شده در برنامه را به صورت خط به خط تشریح خواهیم کرد و تمامی نکات موجود در برنامه را توضیح خواهیم داد تا هیچ نقطه ابهامی باقی نماند و پروژه تمام و کمال در ذهن خواننده مطلب جای بگیرد. در پایان مطالب نیز تمامی کدهای برنامه را به صورت یکجا به شما نمایش خواهیم داد تا بتوانید درک بهتری از این پروژه پیدا کنید.

۲-۱- معرفی کتابخانه های برنامه دفرچه تلفن

در C++ میتوان با دستور `#include` کتابخانه `'import'` کرد .

-اولین کتابخانه ای که ما به آن نیاز داریم :

```
#include <iostream>
```

شما معمولاً در C++ از دستور `cin` و `cout` زیاد استفاده میکنید . دستور `cin` برای دریافت ورودی از سوی کاربر استفاده میشود و شکل کلی این دستور به صورت: `<<نام متغیر>>نام متغیر<<cin` می باشد. دستور `cout` نیز برای خروجی روی مانیتور استفاده می شود و شکل کلی این دستور به صورت: `<<endl<<نام متغیر<<"پیغام"<<cout` می باشد. این دو دستور مهم در C++ در این کتابخانه قرار دارند .

-کتابخانه بعدی که برای نوشتن این برنامه به آن نیاز داریم :

```
#include <stdio.h >
```

برای دستورات کار با فایل مثل `fopen` و... به این کتابخانه نیاز داریم .

-سومین کتابخانه نوشته شده در برنامه :

```
#include <string.h>
```

برای اینکه بتوانیم از رشته ها (`String`) استفاده کنیم باید هدر یا کتابخانه آن را به برنامه متصل کنیم و چون در برنامه ای که نوشتیم به طور مستقیم از رشته ها استفاده کردیم به این کتابخانه نیاز داریم .

-آخرین کتابخانه نوشته شده در برنامه :

```
#include <stdlib.h>
```

۱-import: اصطلاحی کاربردی برای برنامه نویسان به معنی وارد کردن می باشد.

به این کتابخانه برای توابع مربوط به نوشتن دستورات dos نیازمندیم. هر یک از این توابع عملیات خاصی را انجام می دهند مانند تابع `abs (int x)` و ...

۲-۲- نوشتن تابع `main` برنامه

تابع `main` برنامه باید به ما مقدار `int` (عدد صحیح) برگرداند ، مقدار عدد صحیحی که باید به ما برگرداند 0

است پس نوشتن تابع به این صورت می باشد :

```
int main(){  
  
    return 0;  
}
```

نکته: نکته ای که باید به آن توجه داشته باشید این است که اگر برای مثال در خط هفتم برنامه بخواهیم با استفاده از دستور `cout` خروجی در برنامه چاپ کنیم و بنویسیم `cout<<"salam";` و برنامه را اجرا کنیم با ارور مواجه میشویم (ارور آن را در فصل بعدی مشاهده میکنیم) ، این ارور بابت آن است که باید قبل از دستورات `cout` و `cin` از `std::` استفاده کنیم تا برنامه مان اجرا شود یا برای اینکه مجبور نباشیم هر بار قبل از دستورات گفته شده از `std::` استفاده کنیم باید از `using namespace std;` در اول برنامه مان استفاده کنیم (بالای تابع `main`).

```
using namespace std;
```

۱-تابع `main`: تابع `main` اولین تابع از برنامه شما است که زمانی که برنامه شما شروع به اجرا شدن می کند اجرا می شود.

۲-`int`: مخفف `integer` به معنای عدد صحیح می باشد.

۲-۱-۲- حلقه while داخل تابع main

به قطعه کد زیر توجه کنید :

```
int main(){
    int c;
    bool on =true;
    while(on){
        cout<<"Lotfan adadi beyne 0 ta 4 vared konid : \n";
        cout<<"0 - ijade file\n";
        cout<<"1 – gereftane nam va shomare\n";
        cout<<"2 – namayesh liste kamel \n";
        cout<<"3 – jostojoo dar file \n";
        cout<<"4 – khoroj\n";
        cin>>c;
    }
}
```

در مرحله بعد ما می‌خواهیم داخل تابع main برنامه مان حلقه while تعریف کنیم دقت داشته باشید وقتی که شما

می‌خواهید از while,if,swich case و دستورات مانند اینها استفاده کنید بعد آنها از پرانتز، آکولاد و... به صورت

زوج استفاده کنید، یعنی وقتی پرانتز یا آکولادی را باز کردید سریع آن را ببندید که کم یا زیاد نیاید.

در ادامه ما می‌خواهیم حلقه را به صورت بی‌نهایت تعریف کنیم پس ما متغیر on که از نوع بولین^۱ می‌باشد را با

مقدار پیشفرض true تعریف می‌کنیم و این متغیر تا زمانی که ما مقدار خروج را به آن ندادیم (هنگام اجرای این

برنامه با وارد کردن عدد ۴ می‌توانیم از برنامه خارج شویم) قرار است هر دفعه حلقه را اجرا کند و منو را به ما نمایش

دهد.

۱-بولین: مقدار Boolean (در برنامه نویسی bool) بیانگر صحت یا نادرستی شرط یا عبارت است.

در این حلقه قرار است ما ۵ عدد را از کاربر بگیریم و بر حسب آنها کاری را برای کاربر انجام دهیم پس ما متغیری از نوع `int` (عدد صحیح) نیاز داریم که آن را در برنامه به صورت `int c` مینویسیم .

در ادامه ما میخواهیم پیامی را در خروجی چاپ کنیم و به کاربر بگوییم که اعدادی را بین ۰ تا ۴ وارد کند (خط پنجم قطعه کد بالا) که هر کدام از این اعداد کاری را برای کاربر انجام می دهند اما باید مشخص کنیم که هر یک از این اعداد دقیقاً باید چه کاری انجام دهند که از خط ششم تا دهم قطعه کد بالا آن را مشخص کردیم .

نکته: `\n` به معنی این است که هر پیامی را که چاپ می کند به خط بعدی برود .

بعد از اینکه این پیام ها را چاپ کردیم برای کاربر ، کاربر یک سری ورودی به ما می دهد که ما این ورودی ها را داخل متغیر `c` از نوع `integer` (عدد صحیح) میریزیم . (خط یازدهم کد بالا)

۲-۱-۱-۱-۱-۱ دستور switch case

به قطعه کد زیر توجه کنید :

```
switch(c){  
    case 0:  
        break;  
    case 1:  
        break;  
    case 2:  
        break;  
    case 3:  
        break;  
    case 4:
```

```

        break;

        default:

        cout<< "Error!\n";

        break;

    }

```

اگر قصد دارید مقدار یک متغیر منفرد را با استفاده از گزاره‌های متوالی `if..else..if` بررسی کنید، بهتر است به جای آن از گزاره `switch..case` استفاده کنید. همان طور که در خط اول قطعه کد بالا مشخص است جلوی دستور `switch` متغیر `c` تعریف شده است و همانطور که در قسمت های قبل گفتیم این متغیر از نوع `int` (عدد صحیح) می باشد ، دستور `switch` به ما کمک می کند تا این عدد صحیح چاپ شود .

گزاره `break` برای جلوگیری از اجرای کد در `case` بعدی استفاده می شود.

اگر برنامه را اجرا کنیم (اجرای برنامه هم با کلیک بر روی شکلی پنجره مانند داخل محیط `dev c++` امکان پذیر است هم با فشار دادن دکمه `F11`) و برای مثال عدد ۴ را وارد کنیم از برنامه خارج می شود .

در کد دستور شرطی `switch` در `c++` ما از `default` نیز استفاده کردیم. وقتی هیچکدام از شروط اجرا نشود برنامه وارد `default` میشود.

۲-۱-۲-۲- تعریف Error در برنامه

اگر برنامه را اجرا کنیم و به جای عدد از کاراکتر استفاده کنیم حلقه ی بی نهایت به صورت پیاپی خودش را تکرار می کند چون در زبان `c` اگر شما ورودی خود را از نوع `integer` (عدد صحیح) تعریف کرده باشید و کاربر چیز دیگری را وارد کند با ارور مواجه می شوید و شما باید جلوی رخ دادن این ارور را بگیرید . حال برای اینکه ارور های `cin` (دستور ورودی) را خالی کنید باید بعد از اینکه دستور `cin>>c` را که به معنی ورودی از نوع عدد صحیح کاربر بود و در قسمت های قبل توضیح داده شد را تعریف کردیم ،

دستور زیر را بنویسیم :

```
cin.clear();  
while (cin.get() != '\n'){  
    c=5;  
}
```

خط اول قطعه کد بالا به این معنی است که ارور های موجود در cin را clear (پاک) می کند .

در قسمت بعد ما حلقه ی while تعریف می کنیم و در آن از cin.get() استفاده می کنیم که کاراکترها رو از ورودی میگیره و تو متغیر ذخیره می کنه و خط دوم برنامه به این معنی می باشد که تا زمانی که در ورودی اینتر را ندیدیم (\n به معنی خط بعدی برنامه است) و هر چیزی که که کاراکتر خواندیم . و اگر کاربر چیز های نامعقول و کاراکتر وارد کرد با توجه به خط سوم قطعه کد بالا به جای C عدد ۵ را جای گذاری کند .

نکته: != در C++ به معنی نامساوی می باشد.

پس از مراحل بالا برنامه به حالت default می رود (همان طور که در قسمت قبل قطعه کد آن را مشاهده کردید) و بدین ترتیب اگر کاربر کاراکتری را وارد کند برنامه پیام Error را چاپ می کند . (نحوه ی رویداد این ارور را در فصل بعدی مشاهده خواهید کرد.)

۲-۳- تعریف 'struct' (ساختار)

به قطعه کد زیر توجه کنید:

۱- struct : کوتاه شده ی عبارت structure به معنی ساختار می باشد.

```
struct tel{  
  
    char nam[20];  
  
    char shomareh[13];  
  
}d[100];
```

بعضی مواقع نیاز است که شما داده ها با نوع های مختلف را در یک موجودیت گروه بندی کنید و با آن کار کنید. شما با استفاده از ساختار می توانید نوع داده ای خود (که متناسب با صورت مساله و نیاز شماست) تعریف کنید در واقع ساختار مجموعه ای از متغیرها تحت یک نام مشخص است که این متغیرها می توانند نوع داده ای متفاوتی داشته باشند که هر کدامشان با یک اسم مشخص قابل دسترس هستند.

با توجه به مفهوم ساختار ، قطعه کد بالا به این معنی است که هر فردی هم نام و هم شماره تلفن دارد و برای مثال تعیین کردیم که تا ۲۰ کاراکتر کاربر مجاز است که نام خود را وارد کند و تا ۱۳ کاراکتر مجاز است که شماره تلفن خود را وارد کند .

خط چهارم قطعه کد بالا نیز به معنی تعریف آرایه می باشد یعنی در کل ۱۰۰ نفر مجاز هستند که اطلاعات خود را وارد کنند .

۲-۴- نوشتن تابع برای ایجاد فایل

در قسمت های قبل ما یک منو برای برنامه دفترچه تلفن خود تعریف کردیم در ادامه نیاز داریم که برای هر کدام از کارهایی که کاربر می خواهد با وارد کردن اعداد ۰ تا ۴ انجام دهد یک تابع تعریف کنیم.

چون ما می خواهیم در فایل چیزی را ذخیره کنیم باید قطعه کد زیر را در برنامه مان لحاظ کنیم :

```
FILE*fp;
```

به این معنی که متغیر fp از نوع فایل تعریف شده و برای کار با فایل حتما باید این قطعه کد را بنویسیم .

اولین کاری که می‌خواهیم انجام دهیم ایجاد فایل است و آن را در ابتدا به قسمت دستور switch..case که در

قسمت های قبل توضیح داده شد اضافه می کنیم :

```
switch(c){  
    case 0:  
        ijade_file();  
        break;  
}
```

در ادامه تابع زیر را تعریف می کنیم :

```
void ijade_file(){  
    fp=fopen("tel.txt","w");  
    fclose(fp);  
    printf("file ijad shod.\n");  
}
```

در قسمت بالا تابعی مشاهده می شود که خودمان تعریف کردیم ، اینگونه توابع را هم بالای تابع main و هم

پایین آن می توان تعریف کرد ولی تفاوت آنها این است که اگر بخواهیم کل تابع را بالای تابع main تعریف

کنیم نیازی به declaration (اعلان) نداریم در غیر این صورت باید خط اول کد قطعه کد بالا را در بالای تابع

main بنویسیم و باقی تابع را به پایین main منتقل کنیم ولی اگر این کار را انجام ندهیم و کل تابع را به پایین

منتقل کنیم با ارور مواجه می شویم (نحوه رویداد ارور را در فصل بعدی مشاهده خواهید کرد.) زیرا کامپایلر C++

خط به خط برنامه را اسکن می کند و هنگامی که به کد ijade_file() در دستور switch..case می رسد آن

را عبارت نامفهومی تعبیر می کند چون در قسمت های قبل از `ijade_file()` صحبت نکردیم و جزء توابع `c++` هم نیست .

در خط دوم قطعه کد بالا `fp` همان متغیر فایلی که تعریف کرده بودیم برابر با دستور `fopen` است که یک دستور کار با فایل است توجه داشته باشید که برای دستورات کار با فایل از `f` در ابتدای آن ها استفاده می شود . دستور `fopen` به معنی این است که فایلی را می خواهیم ایجاد کنیم یا باز کنیم و اطلاعاتی را به آنها اضافه کنیم . `tel.txt` در قطعه کد بالا به معنی این است که ما می خواهیم فایلی با این نام ایجاد شود . "`w`" به معنی `mode` فایل می باشد (در تابع `fopen` آرگومان `mode` مشخص میکند که فایل باید چگونه باز شود). مود `w` فایلی از نوع `text` را به عنوان نوشتنی باز میکند و مشخص می کند که فایل وجود دارد یا خیر ، اگر وجود داشت آن را خالی می کند و اگر وجود نداشت آن را ایجاد می کند .

با توجه به خط سوم کد بالا تابع `fclose` باعث بسته شدن فایل می شود و شما هنگامی که فایلی را باز می کنید باید آن را ببندید ولی اگر این کار را انجام ندهید تا زمانی که کل برنامه را نبستید اطلاعات `save` (ذخیره) نمی شود .

در خط چهارم قطعه کد بالا دستور `printf` را مشاهده می کنید ، این دستور به معنای دستور `cout` می باشد یعنی ما می خواهیم چیزی را در خروجی چاپ کنیم . با توجه به قطعه کد بالا عبارت `file ijad shod` در خروجی چاپ می شود .

(در فصل آینده ایجاد فایل `tel.txt` را پس از نوشتن کدهای فوق مشاهده خواهید کرد.)

چند نمونه از دستورات معدل هم و کاربردی در C++ :

cout
printf
puts

این ۳ دستور معدل هم هستند و چیزی را در خروجی چاپ می کنند .

cin
scanf
gets

این ۳ دستور نیز معدل هم هستند و به برنامه ورودی می دهند .

برای دستورات کار با فایل از f در ابتدای آن ها استفاده می شود :

fscanf
fgets
fread

این ۳ دستور معدل هم هستند و برای خواندن در فایل به کار می روند.

fprintf
fputs
fwrite

این ۳ دستور نیز معدل هم هستند و برای نوشتن در فایل به کار می روند.

۲-۵- نوشتن تابع برای گرفتن اطلاعات (نام و شماره تلفن) از کاربر

به علت اینکه ما می خواهیم اطلاعاتی را از کاربر بگیریم در ابتدای کار در دستور switch..case به gereftan اشاره می کنیم . چون ما می خواهیم اطلاعات گرفته شده از کاربر را در آرایه ذخیره کنیم متغیری به اسم i از نوع integer (عدد صحیح) نیز تعریف می کنیم و مقدار اولیه آن را 0 می گذاریم . هر دفعه اطلاعات گرفته شده از کاربر در خانه i ام آرایه struct ذخیره می شود و بعد یکی به i اضافه می شود (i++).

در نتیجه به تابع main برنامه مان این قسمت اضافه می شود :

```
int c , i=0;
```

دستور switch..case:

```
switch(c){  
    case 1:  
        gereftan(i++);  
        break;  
}
```

تابع void^۱ برای گرفتن و ذخیره اطلاعات کاربر :

۱-تابع void : تابعی است که در آن مقداری برگردانده نمی شود.

```

Void gereftan (int i){

cout<<"Nam:\n";

cin>>d[i].nam;

cout<<"Shomare \n";

cin>>d[i].shomareh;

fp=fopen("tel.txt" , "a");

fprintf(fp, "%s \t %s \n",d[i].nam,d[i].shomareh);

fclose(fp);

printf( "ezafe shod.\n" );

}

```

همان طور در خط اول قطعه کد بالا مشخص است ورودی ما از نوع `int` (عدد صحیح) می باشد. خط دوم و چهارم قطعه کد بالا نیز به معنی این است که به کاربر می گوید نام و شماره تلفن خود را وارد کند. خط سوم و پنجم آن نیز به این مطلب اشاره دارد که ورودی که به برنامه می دهیم را در خانه `i` ام آرایه `d` ذخیره کند.

با توجه به وجود دستور `fopen` در خط ششم ما می خواهیم فایل را در حالت نوشتن باز کنیم ولی این دفعه `mode` فایل `a` می باشد، این مود فایلی از نوع `text` را طوری باز میکند که بتوان اطلاعاتی را به انتهای آن اضافه نمود (نوشتنی) یعنی هر دفعه می خواهیم اطلاعاتی را به آن اضافه کنیم به طوری که اطلاعات قبلی پاک نشود.

تا به حال فقط فایل را باز کردیم و اطلاعاتی را داخل آن نوشتیم با توجه به خط هفتم قطعه کد بالا با استفاده از دستور `fprintf` می خواهیم چیزی را در فایلمان بنویسیم. در فایل نام و شماره تلفن را که در خانه `i` ام آرایه `d` ذخیره می شوند را می نویسیم. دقت کنید که ما نام و شماره تلفن را از نوع کاراکتری تعریف کردیم، اگر تک کاراکتری باشند می بایست از `%c` استفاده کنیم ولی اگر به صورت رشته یا `string` باشند باید از `%s` استفاده شود.

نکته: کاراکتر «جدول‌بندی» (Tab): این کاراکتر، به وسیله (t\) نمایش داده می‌شود.

در نهایت با توجه به خط هشتم کد بالا فایل را می‌بندیم و با توجه به خط نهم قطعه کد عبارت ezafe shod را در خروجی چاپ می‌کنیم.

۲-۶- نوشتن تابع برای لیست کردن اطلاعات کاربر

همان‌طور که در قسمت‌های قبل گفته شد در ابتدا در دستور switch..case به تابعی که می‌خواهیم بنویسیم اشاره می‌کنیم:

```
switch(c){  
    case 2:  
        list();  
        break;  
}
```

حال به قطعه کد زیر توجه کنید:

```
void list(){  
    char mystring[34];  
    fp=fopen("tel.txt" , "r" );  
    system( "cls" );  
    int f=0;
```



```

puts(" .....\\n");

while(!feof(fp)){

    if(fgets(mystring,34,fp) !=NULL){

        f=1;

        puts(mystring);

    }

}

if(!f)

    puts("File Khali ast.");

puts(" .....\\n");

fclose(fp);

}

```

با توجه به خط دوم کد بالا چون ما می خواهیم خط به خط اطلاعات را بخوانیم کاراکتر `mystring` را تعریف می کنیم علت اینکه برای این کاراکتر عدد ۳۴ در نظر گرفته شده این است که ما ۲۰ کاراکتر برای نام و ۱۳ کاراکتر برای شماره در نظر گرفته بودیم و با احتساب بک اسلش عدد ۳۴ را در نظر گرفتیم .

در خط سوم برنامه فایل را باز می کنیم با مود `r (mode)` . این مود فایل موجود از نوع `text` را به عنوان خواندنی باز میکند و چون ما فقط می خواهیم اطلاعات را بخوانیم از این `mode` استفاده می کنیم .

با توجه به خط چهارم کد بالا در برنامه نویسی `c` ما از دستورات `dos` هم می توانیم استفاده کنیم .

۱- `read : r (mode)` : برای خواندن اطلاعات به کار می رود.

به علت اینکه خروجی برنامه مان شبیه dos است از دستور cls که متعلق به دستورات dos می باشد و باعث پاک شدن اطلاعات می شود استفاده کردیم . برای این که بتوان از cls استفاده کرد دستور کمکی system را نوشتیم که موجب می شود به زبان c از cls استفاده کند . کتابخانه stdlib.h به همین منظور در ابتدای برنامه تعریف شد که ما بتوانیم از دستور system استفاده کنیم.

در خط پنجم قطعه کد بالا `int f=0` را مشاهده می کنید . این متغیر برای زمانی تعریف شده که فایل خالی باشد. همان طور که مشخص است خط ششم قطعه کد بالا مربوط به چاپ کردن می باشد ، یعنی ما می خواهیم نقطه ها (.....) را چاپ کنیم و به خط بعدی برویم چون در زمان اجرای برنامه نظم و ترتیب ایجاد شود و مرز های مشخصی ایجاد شود .

در ادامه (خط هفتم) حلقه ای تعریف شده به این معنی که تا زمانی که به انتهای فایل نرسیدیم . در ادامه آن (خط هشتم) شرطی تعیین شده مبنی بر آن که یک خط از متغیر fp را بخوان (تابع fgetc برای خواندن خط به خط استفاده شده است.) و آن را در mystring قرار بده ، این حلقه تا زمانی ادامه پیدا می کند که به انتهای فایل برسیم . عبارت مخالف NULL نیز به این معنا می باشد که سطری (mystring) وجود دارد. در ادامه (خط نهم کد بالا) `f=1` را مشاهده می کنید به این معنی که سطری وجود داشته که f برابر با ۱ شده است. در خط دهم کد بالا نیز mystring چاپ می شود .

در ادامه نیز (خط سیزدهم و چهاردهم کد بالا) شرطی مشاهده می شد (`if (!f)`) یعنی اگر همچنان برابر با ۰ است . در این شرط اگر f برابر با ۰ باشد ، not (مخالف) آن ۱ می شود و اجرا نمی شود پس شرط if زمانی اجرا می شود که فایل ما خالی باشد در خط بعدی نیز عبارت File Khali ast چاپ می شود .

در خط پانزدهم نیز مجدداً نقطه ها برای ایجاد مرز مشخص (بین کار انجام شده و نمایش مجدد منو) چاپ می شوند و با توجه به کد نوشته شده در خط بعدی فایل ما بسته می شود .

۲-۷- نوشتن تابع برای جستجو کردن اطلاعات

در ابتدا در دستور switch..case به این تابع اشاره می کنیم:

```
switch (c) {  
  
    case 3:  
  
        jostojoo();  
  
        break;  
  
}
```

تابع جستجو با تابع لیست خیلی شباهت دارد بنابراین قسمت هایی که به تابع لیست افزوده شده را با رنگ قرمز

نشان می دهیم و توضیح می دهیم :

```
void jostojoo(){  
  
    char mystring[34],tmp[34], *ch;  
  
    cout<<" Lotfan nam ya shomare ra vared konid: \n";  
  
    cin>>tmp;  
  
    fp=fopen("tel.txt" , "r" );  
  
    system( "cls" );  
  
    int f=0;  
  
    puts(" ..... \n");  
  
    while(!feof(fp)){  
  
        if(fgets(mystring,34,fp) !=NULL){
```

```

        ch = strstr(mystring,tmp);

        if(ch){

            f=1;

            puts(mystring);

        }

    }

    if(!f)

        puts("chizi peyda nashod.");

        puts(" .....\\n");

        fclose(fp);

}

```

در خط دوم قطعه کد بالا tmp[34] را تعریف کردیم که به معنی چیزی است که کاربر وارد کرده است. در خط سوم کد بالا نیز پیام Lotfan nam ya shomare ra vared konid چاپ می شود و با توجه به خط چهارم چیزی که کاربر وارد می کند را به عنوان ورودی می گیریم .

در خط یازدهم کد بالا تابع strstr() را مشاهده می کنیم که برای مقایسه دو رشته mystring (سطری که در آن قرار داریم) و tmp (چیزی که کاربر وارد کرده) به کار می رود، خروجی آن ها را در ch قرار می دهیم که اشاره گر است و آن را به صورت *ch¹ نمایش می دهیم (خط دوم کد بالا). *ch اشاره به کاراکتر دارد .

در ادامه پس از آن که دو رشته را مقایسه کردیم اگر tmp (چیزی که کاربر وارد کرده) زیر مجموعه ای از mystring بود یعنی ch مقدار غیر 0 دارد پس با توجه به خط دوازدهم قطعه کد شرطی تعیین می کنیم که اگر مقدار غیر صفر داشتیم آن زمان mystring را چاپ کند .

در ادامه (با توجه به خط هفدهم و هجدهم) اگر ch مقدار غیر صفر نداشته باشد عبارت chizi peyda nashod را برای ما چاپ کند.

۲-۸- نمایش تمام کد های دفترچه تلفن به صورت یکجا

برای درک بهتر این پروژه بهتر است تمامی کد های آن را به صورت یکجا و زیر هم مشاهده نمایید :

```
#include <iostream>
#include <stdio.h >
#include <string.h>
#include <stdlib.h>
using namespace std;
struct tel{
    char nam[20];
    char shomareh[13];
}d[100];
FILE *fp;
void ijade_file(){
    fp=fopen("tel.txt","w");
    fclose(fp);
    printf("file ijad shod.\n");
```

```

}
Void gereftan (int i){
cout<<"Nam:\n";
cin>>d[i].nam;
cout<<"Shomareh \n";
cin>>d[i].shomareh;
fp=fopen("tel.txt" , "a");
fprintf(fp, "%s \t %s \n" ,d[i].nam,d[i].shomareh);
fclose(fp);
printf( "ezafe shod.\n" );
}
void list(){
    char mystring[34];
    fp=fopen("tel.txt" , "r" );
    system( "cls" );
    int f=0;
    puts(" ..... \n");
    while(!feof(fp)){
        if(fgets(mystring,34,fp) !=NULL){
            f=1;
            puts(mystring);
        }
    }
    if(!f)
        puts("File Khali ast.");
    puts(" ..... \n");
    fclose(fp);
}

```

```

}

void jostojoo(){
    char mystring[34],tmp[34], *ch;
    cout<<" Lotfan nam ya shomare ra vared konid: \n";
    cin>>tmp;
    fp=fopen("tel.txt" , "r" );
    system( "cls" );
    int f=0;
    puts(" ..... \n");
    while(!feof(fp)){
        if(fgets(mystring,34,fp) !=NULL){
            ch = strstr(mystring,tmp);
            if(ch){
                f=1;
                puts(mystring);
            }
        }
    }
    if(!f)
        puts("chizi peyda nashod.");
    puts(" ..... \n");
    fclose(fp);
}

int main(){
    int c,i=0;
    bool on =true;
    while(on){

```

```

    cout<<"Lotfan adadi beyne 0 ta 4 vared konid : \n";
    cout<<"0 - ijade file\n";
    cout<< "1 – gereftane nam va shomare\n";
    cout<< "2 – namayesh liste kamel \n";
    cout<< "3 – jostojoo dar file \n" ;
    cout<< "4 – khoroj\n";

    cin>>c;

    cin.clear();
    while (cin.get() != '\n'){
        c=5;
    }
    switch(c){
        case 0:
            ijade_file();
            break;
        case 1:
            gereftan(i++);
            break;
        case 2:
            list();
            break;
        case 3:
            jostojoo();
            break;
        case 4:
            on = false;
            break;
        default:
            cout<<"Error!\n";

```



```

        break;
    }
return 0;
}
}

```

۲-۹- جمع بندی فصل دوم

در این فصل تمامی کد های برنامه دفترچه تلفن به صورت دقیق به شما شرح داده شد . اولین بخش از این کد ها شامل کتابخانه های این پروژه بود که با `#include` تعریف می شدند . در ادامه با تابع `main` برنامه مان آشنا شدیم که این تابع شامل قسمت های مختلفی از جمله حلقه ی `while` ، دستور `switch..case` و ... بود که به ما کمک کرد تا منوی برنامه مان را بسازیم . در بخش های بعدی این کد ها با `struct` (ساختار) برنامه آشنا شدیم.

در انتهای کار توابع نوشته شده توسط خودمان را که شامل تابع ایجاد فایل ، تابع گرفتن اطلاعات از کاربر ، تابع لیست کردن اطلاعات و تابع جستجو کردن اطلاعات بود برای انجام کارها در دفترچه تلفن به صورت جزئی توضیح دادیم .

فصل سوم

مقدمه

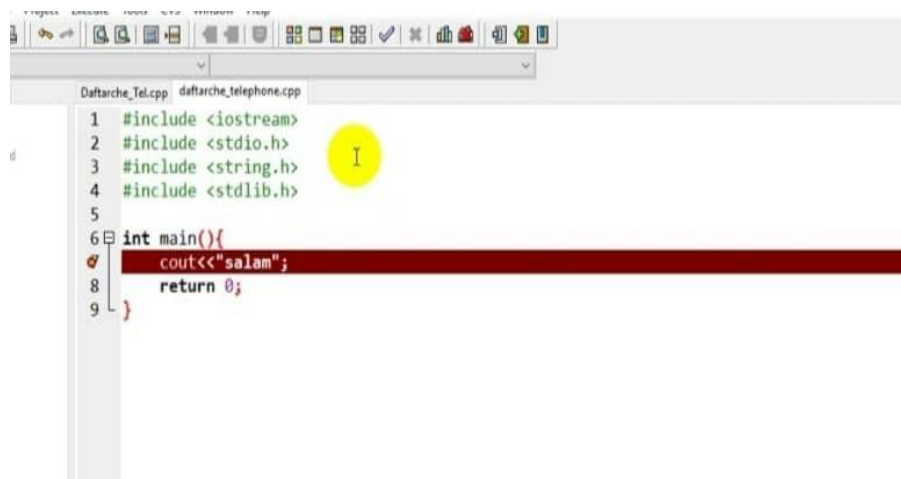
در فصل گذشته ما با تمامی خطوط نوشته شده در برنامه دفترچه تلفن به صورت جزئی و دقیق آشنا شدیم. در این فصل قصد داریم نحوه اجرای برنامه هنگام نوشتن کدها و چگونگی رخ دادن ارور ها را به صورت تصویری برای شما عزیزان نمایش دهیم تا با محیطی که در آن برنامه نویسی کردیم بیشتر آشنا شوید و با مشاهده نحوه کارکرد و اجرای برنامه به درک بیشتری از این پروژه دست یابید. همچنین با کاربرد و اهمیت مهندسی نرم افزار در شکل گیری پروژه آشنا خواهید شد.

۳-۱- نمایش نحوه اجرای برنامه

در این قسمت از این فصل اجرا گرفتن از برنامه دفترچه تلفن طراحی شده را مشاهده خواهید کرد با دکمه F11 می توانید از برنامه اجرا بگیرید :

۱- ارور صورت گرفته به علت استفاده نکردن از دستور `using namespace std` بالای تابع `main` یا

استفاده نکردن از `std::` قبل از دستور `cout` :



شکل ۳-۱

۲- نحوه ی تکرار حلقه ی بی نهایت هنگام وارد کردن "کاراکتر" در زمانی که ما فقط ورودی را از نوع `int` (عدد صحیح) تعریف کردیم و تابعی تعریف نکردیم که ارور های موجود در `cin` را پاک (clear) کند:


```

1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
1
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
2
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
3
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
4
-----
Process exited with return value 0
Press any key to continue . . .

```

شکل ۳-۴

۴- نحوه ی رخ دادن ارور هنگامی که توابعی را خودمان تعریف کردیم و آن ها را بدون اینکه اعلان کنیم در پایین تابع main نوشتیم: (در تصویر زیر تابع ایجاد فایل را داریم که در پایین تابع main نوشته شده است)

```

29
30 switch(c){
31     case 0:
32         ijade_file();
33         break;
34     case 1:
35         break;
36     case 2:
37         break;
38     case 3:
39         break;
40     case 4:
41         on = false;
42         break;
43     default:
44         cout<<"Error!\n";
45         break;
46 }
47 return 0;
48
49
50
51 void ijade_file(){
52     fp = fopen("tel.txt","w");
53     fclose(fp);
54     printf("file ijad shod.\n");
55 }

```

شکل ۳-۵

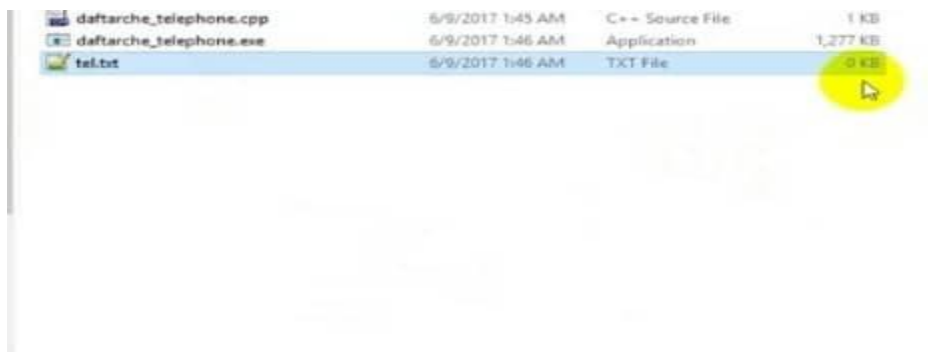
۵- نحوه ی اجرای تابع ijade_file() پس از وارد کردن عدد 0 و ایجاد فایلی از نوع TXT:

```

Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
0
file ijad shod.
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj

```

شکل ۳-۶



شکل ۳-۷

۶- نحوه ی اجرای تابع `gereftan(i++)` با وارد کردن عدد ۱ (گرفتن نام و شماره تلفن فرد) و ذخیره اطلاعات در فایل TXT:

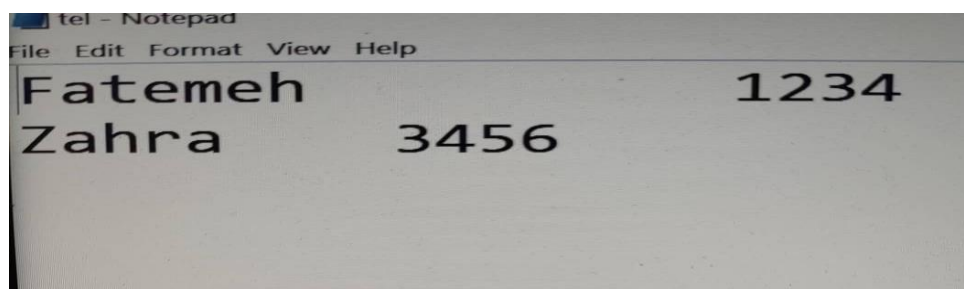
نکته: دقت داشته باشید که اگر اطلاعاتی در فایل TXT موجود باشد و شما بخواهید اطلاعات جدیدی را وارد کنید و اطلاعات قبلی را پاک کنید می توانید با وارد کردن عدد ۰ این کار را انجام دهید.

```

Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
1
Nam:
Fatemeh
shomare:
1234
ezafe shod.
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
1
Nam:
Zahra
shomare:
3456
ezafe shod.
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
4

```

شکل ۳-۸



شکل ۳-۹

۷- نحوه ی اجرای تابع `list()` با وارد کردن عدد ۲ و لیست کردن تمام اطلاعات موجود در فایل:

نکته: دقت داشته باشید که اگر فایل خالی باشد برای ما `File Khali ast` را چاپ می کند.

```
.....
Fatemeh          1234
Zahra            3456
.....
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kame1
3 - jostojoo dar file
4 - khoroj
_
```

شکل ۳-۱۰

```
.....
File Khali ast.
.....
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kame1
3 - jostojoo dar file
4 - khoroj
```

شکل ۳-۱۱

۸- نحوه ی اجرای تابع `jostojoo()` با وارد کردن عدد ۳ (شما می توانید با وارد کردن بخشی از نام یا شماره

تلفن ، اطلاعات کاربرانی را که شامل آن بخش هستند را مشاهده کنید):

برای مثال در تصاویر زیر عدد ۳۴ برای جستجو کردن وارد کردیم که برنامه اطلاعات ۲ نفر را که شماره تلفنشان شامل این عدد بود را به ما نشان داد.

```

Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj
3
Lotfan nam ya shomare ra vared konid:
34

```

```

.....
FatemeH          1234
Zahra           3456
.....
Lotfan adadi beyne 0 ta 4 ra vared konid:
0 - ijade file
1 - gereftane nam va shomare
2 - namayesh liste kamel
3 - jostojoo dar file
4 - khoroj

```

شکل ۳-۱۲

۳-۲- کاربرد اصول مهندسی نرم افزار در طراحی دفترچه تلفن

مهندسی نرم افزار کاربرد یک روش سیستماتیک، علمی و کمیت پذیر در بسط، راه اندازی و نگهداری نرم افزار

می باشد اما اگر اصول مهندسی رعایت نشود انجام چنین عملی امکان پذیر نمی باشد .

این اصول، الگوها، طرح ها و روش های تولید نرم افزار خوب، هسته اصلی مهندسی نرم افزار را تشکیل می دهد و

توجه به آنها در مهندسی نرم افزار بسیار ضروری می باشد .

به طور کلی این اصول شامل ۷ اصل می باشند :

۱-دلیل وجود سیستم ۲-ساده نگه داشتن ۳- حفظ چشم انداز ۴-آنچه که شما تولید می کنید، دیگران مصرف می کنند ۵-آینده نگری ۶-برنامه ریزی پیشاپیش برای استفاده مجدد ۷-تفکر

حال چگونه این اصول در پروژه ی ما مؤثر بودند؟ همان طور که می دانید پروژه ی دفترچه تلفن که به زبان ++C برای شما توضیح داده شد پروژه ی بسیار ساده ای است و طراحی ها در آن بسیار ساده هستند و برای کسانی که ابتدای راه یادگیری زبان سی پلاس پلاس هستند بسیار کاربردی و مفید است ، همین ویژگی در این پروژه باعث می شود که سیستم قابل فهم تری داشته باشم (اصل ساده نگه داشتن). علت وجود پروژه ی دفترچه تلفن ایجاد ارزش برای کاربران می باشد و باعث می شود که کاربر بتواند اطلاعاتش را در آن ذخیره کند و جستجوی سریعی داشته باشد(دلیل وجود سیستم). پروژه ی طراحی شده دارای یکپارچگی مفهومی است و همواره تعیین مشخصات، طراحی و پیاده سازی در آن طوری انجام شده که دیگران نیز قادر به درک کار ما باشند (حفظ چشم انداز - آن چه که شما تولید می کنید دیگران مصرف می کنند). ایجاد برنامه ای با طول عمر بالا از ارزش بیشتری برخوردار است پس آن را طوری طراحی کردیم که آمادگی انطباق بر تغییرات را داشته باشد و بتوانیم کد های جدید نیز به آن اضافه کنیم (آینده نگری). تعقل و تفکر کامل و روشن قبل از اقدام به عمل، همواره نتایج بهتری به بار می آورد و همان طور که واضح است بدون تفکر، ساخت و طراحی پروژه با اختلال مواجه می شود و توجه به این اصل باعث بالا رفتن ارزش پروژه ی طراحی شده می شود (تفکر).

۳-۳- کاربرد فرایند های مهندسی نرم افزار در طراحی پروژه

هر پروژه ی نرم افزاری، چه بزرگ و چه کوچک مراحل را طی می نماید که در طی آن مجموعه ای از نیازمندی های مشتری به یک محصول نرم افزاری تبدیل می گردد. الگو و قالبی که چگونگی طی مراحل مختلف یک پروژه را تعریف می نماید، اصطلاحاً فرآیند تولید نرم افزار نامیده می شود.

حال فرایند های مهندسی نرم افزار شامل ارتباطات^۱ - برنامه ریزی^۲ - مدل سازی^۳ - ساخت^۴ - استقرار^۵ می شوند.

چگونه این فرایند ها به ایجاد پروژه ی ما کمک کرده اند؟ برای اینکه پروژه ای طراحی شود نیاز است که با افراد و اشخاص دیگر ارتباط بگیریم ، نیاز های آن ها را بسنجیم و خواسته های آن ها را جمع آوری کنیم انجام این اعمال باعث می شود که ویژگی ها و قابلیت های عملیاتی نرم افزار را بتوانیم تعیین کنیم .طراحی پروژه ما نیز به همین صورت بوده و ما بر اساس شناخت نیاز های دیگران و کارهایی که با طراحی دفترچه تلفن قصد انجام آن را داشتند ، قابلیت هایی که باید این پروژه شامل آن می شد را تعیین کردیم(ارتباطات). با برنامه ریزی ، زمانبندی کاری و گذاشتن زمانی مشخص برای تعریف هر یک از توابع پروژه ، طراحی این پروژه به سهولت صورت گرفت (برنامه ریزی). برای اینکه ما بتوانیم خواسته ها را بهتر درک کنیم و آن ها را طراحی کنیم نیازمند مدلی هستیم که در ابتدای کار ، عملی که می خواهیم انجام بدهیم را به ما نشان دهد .پس ما در ابتدا منوی دفترچه و کارهایی را که با زدن اعداد منو نیازمند انجام آن ها بودیم را در کاغذ مدل سازی کردیم (مدل سازی) و بر طبق آن ها شروع به نوشتن برنامه نویسی کردیم ، کد ها را تولید کردیم و برای آشکار کردن خطاهای موجود ،آزمون های لازم را انجام دادیم (ساخت). در پایان کار نیز برنامه ای را که تولید کردیم برای ارزیابی به دیگران تحویل دادیم تا بر اساس آن ارزیابی بازخوردی را به ما ارائه کنند (استقرار).

Communication-۱

Planning-۲

Modeling-۳

Construction-۴

Deployment-۵

۳-۴- چگونه دانش مهندسی نرم افزار به طراحی پروژه کمک نمود ؟

برای اینکه پروژه دفترچه تلفن را طراحی کنیم در ابتدای کار تحلیل و بررسی کردیم که چه کاری باید انجام دهیم ، نیازهای اشخاص دیگر را شناختیم و براساس آن ها شروع به طراحی پروژه ی مورد نظر کردیم و یک مدل از آن را طراحی کردیم و سپس شروع به ساخت پروژه مان کردیم و کد نویسی های لازم را انجام دادیم . در انتهای کار نیز برنامه ای که ساخته بودیم را برای تأیید به دیگران تحویل دادیم .

در واقع تمامی کارهایی که ما انجام دادیم ، اعمال مهندسی محسوب می شوند زیرا به طور کلی مهندسی عبارت است از : تحلیل - طراحی - ساخت - تأیید - مدیریت امور فنی

در نتیجه کاملاً روشن و واضح است که بدون در اختیار داشتن دانش مهندسی نرم افزار ، ساخت چنین پروژه ای غیر ممکن بود .

کارهای مهندسی نرم افزار بدون توجه به حوزه ی کاربرد ، اندازه ی پروژه و پیچیدگی آن به ۳ گروه کلی تقسیم می شود و به سرانجام رسیدن پروژه ی ما نیز مستلزم توجه به این ۳ گروه بود .

مرحله ی تعریف : این مرحله روی آنچه که کار ها هستند (what) متمرکز می شود یعنی در این مرحله متوجه این موضوع شدیم که چه نوع اطلاعاتی را باید در پروژه مان به کار ببریم ، در پروژه چه نوع رابط هایی باید ایجاد شوند ، محدودیت های پروژه مان چیست و سیستم مورد انتظارمان چه وضعیتی باید داشته باشد .

در واقع ما در این مرحله مهندسی سیستم یا اطلاعات را انجام دادیم ، پروژه ی نرم افزاری را طراحی کردیم و موارد مورد نیاز را تحلیل کردیم .

مرحله ی توسعه : مرحله ی توسعه و ارائه روی چگونگی (how) متمرکز است یعنی در این مرحله متوجه شدیم که داده های برنامه ما چگونه باید ساخته شوند ، نحوه ی اجرای کار در ساختار نرم افزاری ما به چه صورت است ، چگونه باید طرح به زبان برنامه نویسی تبدیل شود و چگونه آزمون روی پروژه ی ما باید انجام شود .

در نتیجه ما در این مرحله پروژه (نرم افزار) را طراحی کردیم ، کد ها را تولید کردیم و آزمون نرم افزار را انجام دادیم .

مرحله ی پشتیبانی : این مرحله روی تغییر (change) متمرکز است به این معنی که ما در این مرحله خطاهای موجود در پروژه مان را اصلاح کردیم و پروژه را به صورتی طراحی کردیم که اگر نیاز های جدیدی مطرح شد بتوانیم آن را تغییر دهیم و کد های جدیدی به آن اضافه یا از آن کم کنیم و نیاز های جدید را رفع کنیم .

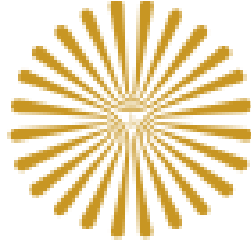
۳-۵- جمع بندی فصل سوم

در این فصل به شما با تصاویر نشان دادیم که پس از کدنویسی کردن ، برنامه به چه صورت اجرا می شود و در چه جاهایی با ارور مواجه می شویم و علت آن چیست . برای مثال نحوه ی نمایش منوی برنامه ، نحوه ی اجرا شدن برنامه پس از تعریف توابع ایجاد فایل ، گرفتن (اطلاعات کاربر) و ... ، ارور برنامه به دلیل استفاده نکردن از دستور `using namespace std` ، ارور برنامه هنگامی که تابعی را خودمان تعریف کردیم و بدون اینکه آن را اعلان کنیم در زیر تابع `main` نوشتیم و ... را به شما نشان دادیم تا با کارکرد دفترچه تلفن آشنا شوید . در پایان مطالب نیز اهمیت مهندسی نرم افزار (دانش ، اصول و فرایندهای مهندسی نرم افزار) در به سرانجام رساندن پروژه را برای شما بیان کردیم .

جمع بندی و ارائه پیشنهادات

گزارش پروژه دفترچه تلفن به زبان C++ را در قالب ۳ فصل به شما عزیزان ارائه کردیم. فصل اول این گزارش در ارتباط با زبان سی پلاس پلاس بود که در آن به مواردی از قبیل تاریخچه این زبان، توابع، متغیرها، عملگرها، محیط‌هایی که می‌شود در آن‌ها این زبان برنامه‌نویسی را کامپایل کرد و نصب یکی از طرفدارترین محیط‌ها که برنامه دفترچه تلفن در آن نوشته شده است، می‌توان اشاره کرد. در فصل دوم این گزارش تمامی کد نویسی‌های پروژه مان را به شما نشان دادیم و به صورت بسیار دقیق و جزئی در ارتباط با آن‌ها توضیح دادیم. در فصل پایانی نیز طریقه اجرا و کارکرد دفترچه تلفن طراحی شده را به شما نشان دادیم و به شما این مطلب را اثبات کردیم که بدون داشتن علم مهندسی و برنامه‌نویسی، در انجام چنین پروژه‌ای ناکام می‌ماندیم.

از آن جایی که زبان C++ یک زبان basic یا پایه‌ای محسوب می‌شود، اگر بتوانیم آن را فرا بگیریم زمینه‌ی لازم برای یادگیری زبان‌های برنامه‌نویسی دیگر را ایجاد کردیم. به همین علت در محیط‌های آموزشی مانند دانشگاه در ابتدا به مهندسين کامپیوتر این زبان را آموزش می‌دهند. پیشنهاد من به شما عزیزان این است که اگر به دنبال یادگیری زبان برنامه‌نویسی C++ هستید، این پروژه‌ی دفترچه تلفن را که خدمت شما توضیح دادیم انجام دهید زیرا برای کسانی که در ابتدای راه هستند بسیار کاربردی و مفید است و در آن از پرکاربردترین و مهم‌ترین توابع و دستورات سی پلاس پلاس استفاده شده است.



**payam noor parand Faculty of
Department of Computer engineering**

B.Sc. Final Project Report

Title of the Report:

C++ Phonebook

Under Supervision of:

d.r Ali Razavi

By:

Fatemeh Akbarshahi

<Date>

October of 2021

