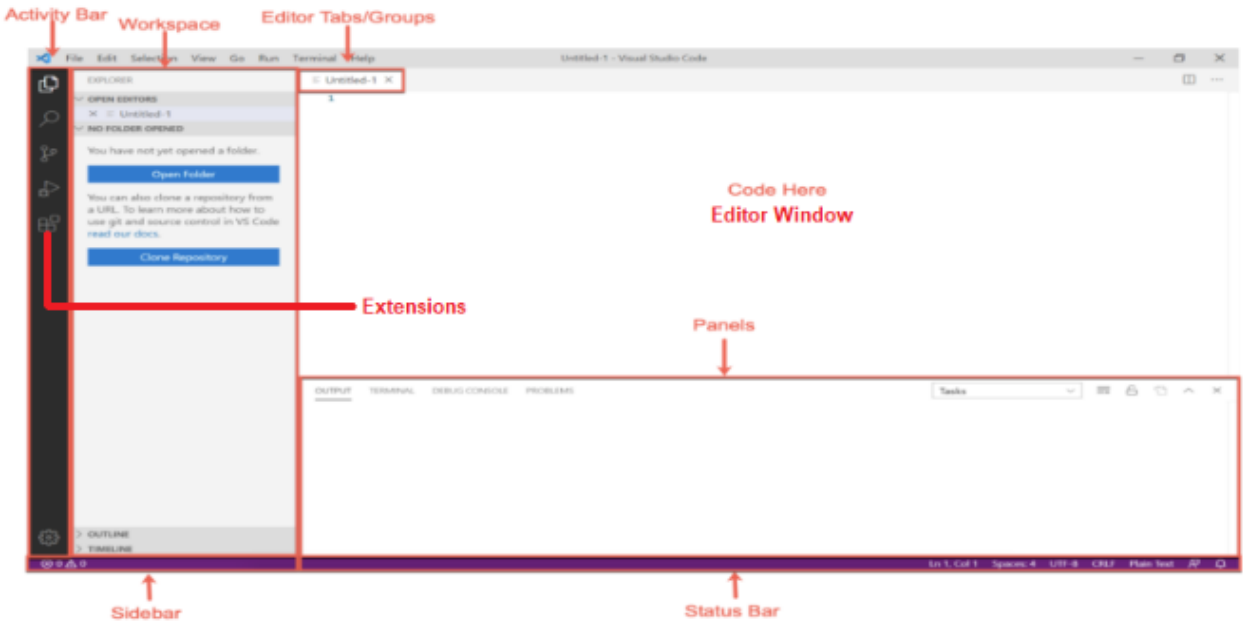
نیمسال اول **1402-03**	عنوان: دستور کار آزمایشگاه شماره 2 آشنایی مقدماتی با C ، متغیرها و کار با ورودی و خروجی	اساتید: دکترمرادی، دکتر هاشمی

در این جلسه شما قرار است تمرین بیش تری در محیط برنامه نویسی VSCode داشته و با نحوه‌ی استفاده از توابع ورودی و خروجی بیش تر آشنا شوید.

پیش نیاز (آشنایی با محیط نرم افزار):

هر گاوچرانی به یک اسب و هر کدنویس به یک ویرایشگر کد عالی نیاز دارد. که ما در این آموزش قرار است VSCode را آموزش دهیم

وقتی برای اولین بار این برنامه را اجرا می‌کنید یک صفحه مانند شکل پایین مشاهده خواهید کرد که اجزای آن را به مرور توضیح خواهیم داد.



Editor Window:

پنجره ویرایشگر جایی است که شما بیشتر کار خود را انجام خواهید داد. در این صفحه می‌توانید تمام کد هایی را که روی آن کار می‌کنید مشاهده و ویرایش کنید . هر زمان که یک فایل جدید را باز می‌کنید یا یک فایل موجود را ویرایش می‌کنید، پنجره ویرایشگر جایی است که کد نمایش داده می‌شود. VScode دارای برگه هایی در این ویرایشگراست که به شما امکان می‌دهد چندین فایل را همزمان باز کنید و گروه‌های ویرایشگر که برگه‌های مختلف را گروه بندی می‌کنند.

Workspace:

رایج ترین بخش بعدی رابط کاربری [1]خواهد بود که استفاده می‌کنید . فضای کاری [2]جایی است که هر فایلی که در برگه‌ها [3]باز کرده اید و همچنان پوشه‌ای که در آن قرار دارید را نشان می‌دهد. از این قسمت می‌توانید دسترسی راحت تری به سایر فایل‌های پروژه داشته باشید.

Panels:

بخش پانل‌ها بخش "خروجی" است. در این بخش «برگه های» مختلفی را با اطلاعاتی که توسط VScode و پسوندهای آن برگردانده شده اند پیدا خواهید کرد. اینجا جایی است که ترمینال را نیز پیدا خواهید کرد.

Status Bar:

در قسمت نوار وضعیت می‌توانید اطلاعات مربوط به پروژه و وضعیت آن مانند تعداد خطاها و هشدارها و همچنین نوع فایل و ... را مشاهده کنید.

Extensions:

یکی از مهم ترین ویژگی‌های وی اس کد، افزونه‌های آن است. VScode از ویژگی‌های رایج ویرایشگر کد مانند برجسته سازی دستور زبان، IntelliSense و غیره پشتیبانی می‌کند. نوع کدی که می‌نویسید را درک می‌کند و به طرق مختلف با آن سازگار می‌شود. اما همه زبان‌ها را نمی فهمد. C فقط یک متن ساده است. برای اینکه VScode بداند چه زمانی یک بخش باید جمع شود و رنگ چه متنی را باید تغییر دهد، باید ساختار C را درک کند. VScode بسته به نوع فایل باز شده، این ویژگی‌ها را متفاوت اعمال می‌کند.

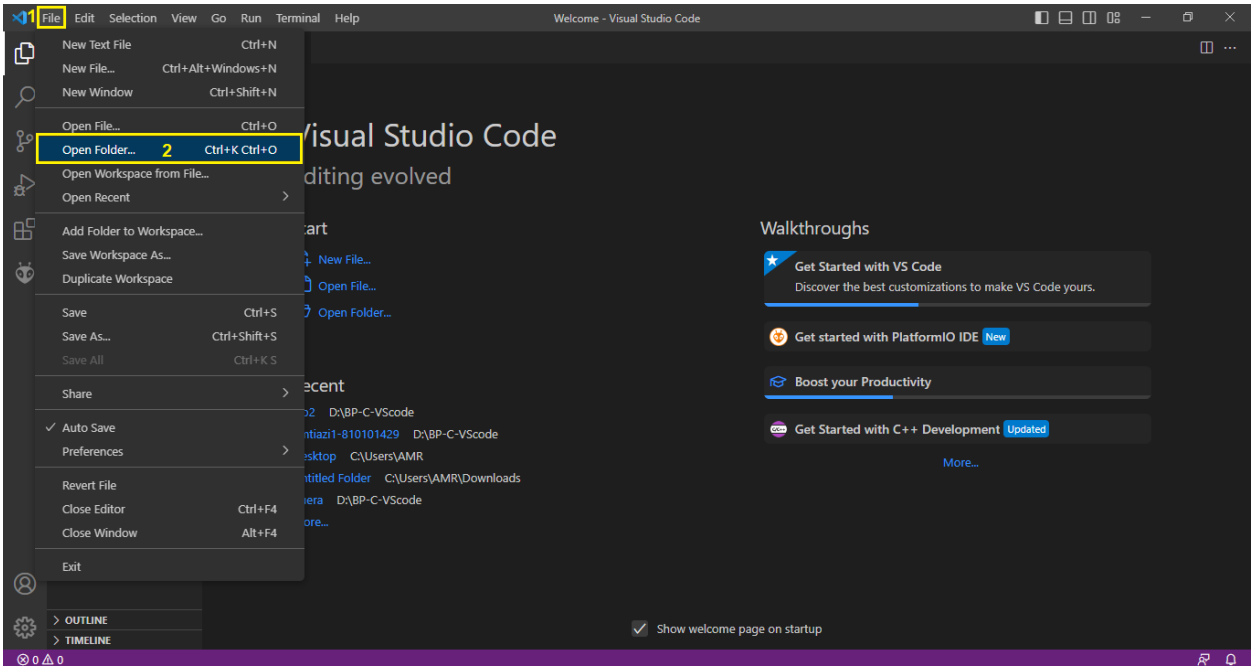
---

[1] User interface
[2] Work space
[3] Tabs

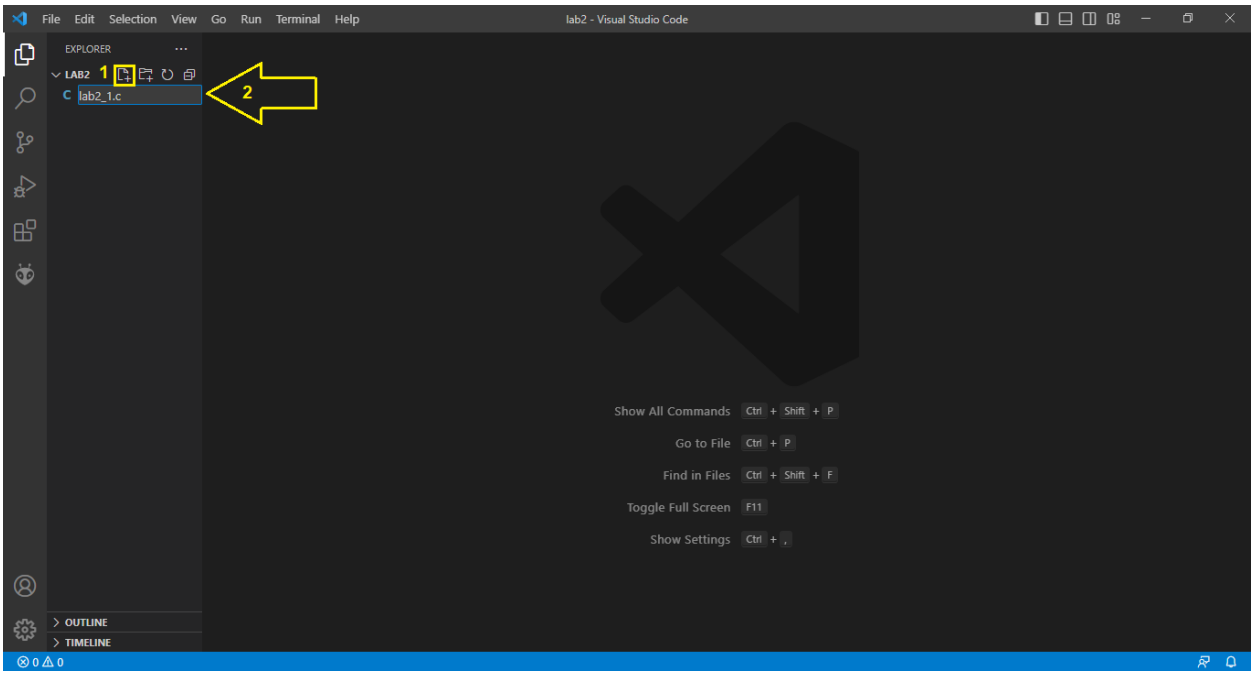
1- انجام دهید. (یک برنامه ساده)

در این قسمت قصد داریم (احتمالا!) اولین برنامه خود را نوشته و در خروجی نتیجه را مشاهده کنیم.

در ابتدا توصیه می‌شود در سیستم خود، یک پوشه برای ذخیره سازی فایل کد ایجاد کنید. حال مطابق تصویر زیر، پس از اجرای VScode، از منوی Folder گزینه Open Folder را انتخاب کرده و سپس پوشه‌ای را که ایجاد کرده اید، انتخاب نمایید.

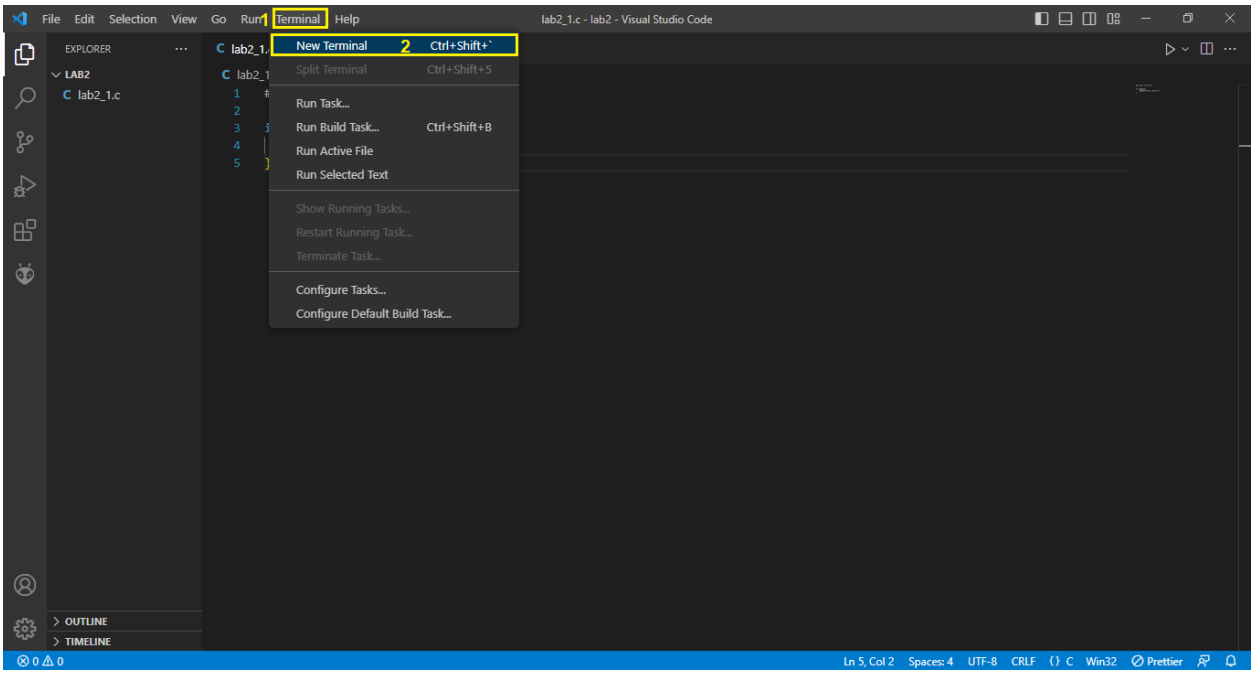


حال که پوشه مورد نظر را باز کردید، نشانگر ماوس خود را روی اسم فایل که در قسمت چپ برنامه قابل مشاهده است برده و روی آیکونی که در تصویر زیر مشخص شده کلیک نمایید. با این کار شما در پوشه خود یک فایل جدید (New File) ایجاد می‌کنید. سپس باید یک نام برای این فایل انتخاب کنید. دقت داشته باشید پس از نوشتن نام فایل حتما در انتها پسوند **.c** را به نام آن اضافه کنید تا به یک فایل زبان C تبدیل گردد. در اینجا نام فایل به صورت lab2_1.c انتخاب شده است.



سپس کد را می‌نویسیم:

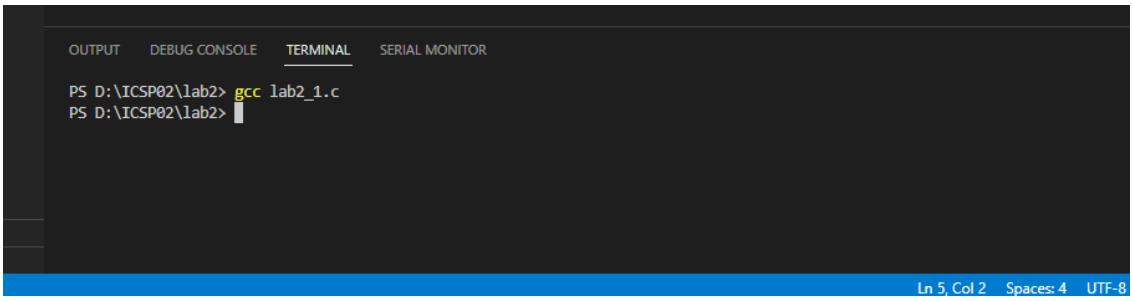
اکنون باید این کد را کامپایل کنیم. بنابراین ابتدا باید ترمینال را باز کنیم. پس ابتدا روی گزینه Terminal کلیک کرده و سپس New Terminal را انتخاب کنید. این کار با فشردن کلیدهای ترکیبی زیر نیز ممکن است: **ctrl + shift + `**



برای کامپایل کردن هر فایلی توسط کامپایلر gcc کافیست دستور زیر را در تر مینال وارد کرده و در نهایت enter را بزنید.

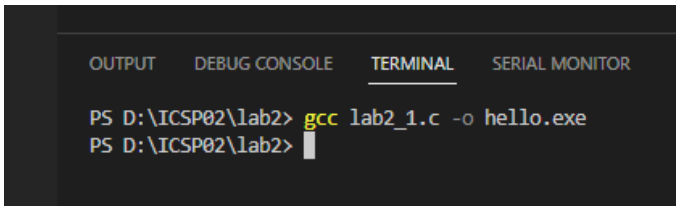
gcc نام فایل

برای مثال ما در اینجا مطابق تصویر زیر فایل خود را کامپایل می‌کنیم:



همانطور که در پنجره کناری می‌بینید، فایلی با نام a.exe به پوشه اضافه شده است که همان فایل اجرایی می‌باشد. در صورت تمایل می‌توانید به جای عبارت بالا، عبارت زیر را در ترمینال وارد کرده و فایل اجرایی با نام دلخواهتان (در اینجا hello.exe) ذخیره گردد.

```
gcc lab2_1.c -o hello.exe
```



برای اجرای هر فایل اجرایی، کافیست در ترمینال عبارت **./** را تایپ کرده و پس از آن نام فایل را وارد نمایید. سپس enter را بزنید. برای مثال در این برنامه به صورت زیر عمل می‌کنیم:

همانطور که در تصویر می‌بینید، عبارت `Hello World!` را نمایش می‌دهد.

✅ قسمت 1: نتایج را به دستیاران آموزشی نشان دهید.

عملیات خواندن از ورودی و نوشتن در خروجی توسط دو تابع scanf و printf :

توابع scanf و printf به ترتیب توابع ورودی و خروجی استاندارد فرمت دار هستند. یعنی شما می‌توانید فرمت داده‌ای را که می‌خواهید بخوانید یا بنویسید را تعیین کنید. برخی فرمت‌ها در جدول زیر آمده است:

Format	Format Specifier
اعداد صحیح دهدهی مثبت و منفی (int)	%d or %i
عدد اعشاری ممیز شناور 32 بیت (float)	%f
عدد اعشاری ممیز شناور 64 بیت (double)	%lf
اعداد مبنای 16 مثبت (hex)	%x
یک کاراکتر (char)	%c
رشته‌ای از کاراکترها (عبارت رشته ای) (string)	%s

2- انجام دهید. (یک برنامه ساده دیگر!)

حال این قطعه کد را در یک پروژه‌ی جدید اجرا کنید تا توضیحات بالا بیشتر برایتان جا بیفتد.

```
#include <stdio.h>
int main(){
    int x, y;
    scanf("%d", &x);
    scanf("%d", &y);
    printf("The result is: %d\n", ((x + y) << 2) % 3);
}
```

حال سعی کنید قطعه کد بالا را طوری تغییر دهید که کاربر بتواند دو عدد ورودی را در یک خط و با یک کاراکتر space بین آن دو وارد کند. می‌توانید برای پیدا کردن پاسخ سوال خود به این لینک مراجعه کنید.

قسمت 2: نتایج را به دستیاران آموزشی نشان دهید.

به نظرتان اگر در تابع `scanf`، در انتهای فرمت عبارت `\n` را قرار دهیم چه اتفاقی می‌افتد؟ علت را برای دستیاران آموزشی توضیح دهید. (قسمت 3)

اکنون در دستور `scanf` ، علامت `&` که قبل از `x` گذاشته شده را بردارید و مجددا برنامه را کامپایل و اجرا نمایید. حال یک عدد را به عنوان ورودی وارد کنید. چه اتفاقی افتاد؟ چرا برنامه از کار افتاد؟ در این مورد به دو نکته زیر توجه کنید:

1- شما علت دقیق بروز این خطا را بعد از آشنایی با مفاهیم تابع[4] و نحوه‌ی آرگومان دهی[5] به آن، اشاره گرها[6] و مرجع‌ها[7] در همین درس خواهید آموخت، ولی در حال حاضر همین قدر بدانید که اگر در تابع scanf علامت & قبل از x را فراموش کنید بگذارید، برنامه‌ی شما می‌خواهد به خانه‌ای از حافظه دسترسی داشته باشد که مال خودش نیست! برای همین سیستم عامل جلوی اجرای برنامه را می‌گیرد.

2- خطایی که با آن مواجه شدید خطایی بود که به هنگام اجرا[8] رخ داد و خطای زمان کامپایل[9] نبود.(زیرا برنامه شما به درستی و بدون خطا کامپایل شد).

---

[4] Function
[5] Argument passing
[6] pointers
[7] References
[8] Runtime Error
[9] Compile Error

حال در یک فایل جدید، قطعه کد زیر را اجرا کرده و نتایج را به ازای ورودی‌های مختلف بررسی کنید.

```
#include <stdio.h>
int main(){
    int x;
    scanf("%d", &x);
    printf("The result is: %d\n", &x);
}
```

با تغییر مقدار ورودی، چه تفاوتی در نتیجه رخ می‌دهد؟ علت را برای دستیاران آموزشی توضیح دهید. (قسمت 4)

خط اول برنامه فوق (یعنی `#include <stdio.h>`) را حذف کنید و مجدداً برنامه را کامپایل نمایید. چه اتفاقی می‌افتد؟ پیغام خطایی که کامپایلر به شما می‌دهد به چه معنا است؟ علت را برای دستیاران آموزشی توضیح دهید. (قسمت 5)

3- انجام دهید!

متغیرها در کامپیوتر به روشهای مختلفی ذخیره میشوند. از این روشها میتوان روش ASCII برای متغیر از نوع char و یا سیستم نمایش Floating Point برای اعدا اعشاری یا متغیر float نام برد.

قطعه کد زیر را در یک پروژه جدید اجرا کنید. سپس به عنوان ورودی کاراکتر S را وارد نمایید.

```
#include <stdio.h>
int main() {
    char x;
    printf("Enter a character:\n");
    scanf("%c", &x);
    printf("%d\n", x);
}
```

حال قطعه کد زیر را اجرا کرده و خروجی کد بالا را به عنوان ورودی وارد کنید.

```
#include <stdio.h>
int main() {
    int x;
    printf("Enter a number:\n");
    scanf("%d", &x);
    printf("%c\n", x);
}
```

عدد مشاهده شده در خروجی کد اول نمایانگر چه مقداری است؟ در کد دوم چه اتفاقی افتاد؟ علت را برای دستیاران آموزشی توضیح دهید.(قسمت 5)

قطعه کد زیر را در یک پروژه جدید اجرا کنید. سپس به عنوان ورودی یک بار عدد 5 و یک بار عدد 1092091904 را وارد کنید.

```
#include <stdio.h>
int main() {
    float x;
    printf("Enter a decimal number:\n");
    scanf("%d", &x);
    printf("%f\n", x);
}
```

اعداد مشاهده شده در خروجی نمایانگر چه مقادیری هستند؟

راهنمایی: مقادیری که در متغیرهای float ذخیره می‌کنیم، در کامپیوتر در سیستم floating point و طبق استاندارد IEEE 754 و به صورت 32 بیتی ذخیره می‌شوند.

حال مقدار عدد مشاهده شده در خروجی به ازای ورودی 1092091904 را در سیستم Floating Point محاسبه کنید. سپس با استفاده ار مبدل اعداد binary به decimal در این لینک مقدار عدد را در مبنای 10 محاسبه کنید. چه نتیجه‌ای می‌گیرید؟ نتایج را برای دستیاران آموزشی توضیح دهید.(قسمت 6)

برای نمایش یک عدد اعشاری تا n رقم اعشار می‌توانید از یکی دیگر از قابلیت‌های تابع printf استفاده کنید! به کد زیر توجه کنید:

```
    float x;
    printf("%.2f\n", x);
```

با این کار می‌توانید عدد اعشاری x را تا 2 رقم اعشار نشان دهید.

حال با توجه به این موضوع قطعه کد قسمت قبل را طوری تغییر دهید که خروجی کد به ازای ورودی 5، صفر نباشد. نتایج را برای دستیاران آموزشی توضیح دهید.(قسمت 7)

فرمت‌ها در دستور printf بسیار پرکاربرد هستند و می‌توانند خروجی شما را خوانا و زیبا کنند. برای آشنایی بیش تر با فرمت‌ها می‌توانید به این لینک مراجعه کنید.

4- انجام دهید!(دایره!)

می خواهیم برنامه‌ای بنویسیم که عدد اعشاری r را از کاربر بگیرد و محیط و مساحت دایره‌ای به شعاع r را با دقت نسبتا بالایی حساب کند ونتیجه را تا 4 رقم اعشار نمایش دهد. مثلا برای حالت r=10 ، خروجی به صورت زیر خواهد بود:

The result is: 314.1593

برنامه زیر را در نظر بگیرید:

```
#include <stdio.h>

int main() {
    double r;
    printf("Please enter the value of r:\n");
    scanf("%...", &r);
    printf("The area is: %...\n", (3.141593*r*r));   /*نمایش مساحت دایره*/
    printf("The circumference is: %...\n", (2*3.141592*r));   /*نمایش محیط دایره*/
    return 0;

}
```

ابتدا در برنامه‌ی بالا، جاهای خالی را که هایلایت شده اند، تکمیل کنید. دلیل انتخابتان را برای دستیاران آموزشی توضیح دهید.(قسمت 8)

حال بیشتر به این برنامه توجه کنید. به نظرتان کد بالا چه ایرادی دارد؟ بله، فردی که کد بالا را نوشته، یکجا عدد π را 3.141593 و در سطر بعدی، آنرا برابر با 3.141592 وارد کرده است. به نظرتان چگونه می‌توان جلوی چنین اشتباهاتی را گرفت؟

حال عبارت `#define PI 3.141593` را در ابتدای کد بالا (پس از include) بنویسید. با این تعریف در هر جای کدتان می‌توانید به جای عدد 3.141593 از برچسبی[10] به نام PI استفاده کنید. اکنون تغییرات بیان شده را روی کد بالا اعمال کنید.

حال نتایج را به دستیاران آموزشی نشان دهید.(قسمت 9)

توجه: یکی از ویژگی‌های یک برنامه‌ی خوب، کاربر پسند[11] بودن آن است. برای رعایت این نکته پیش از خواندن ورودی ابتدا باید به کاربر پیغام مناسب بدهید.

[10] Label
[11] User friendly

5- انجام دهید!

اکنون قطعه کد زیر را اجرا کنید.

```
#include <stdio.h>
int main() {
    float x = 0.3;
    printf("%.30f\n", x + x + x + x + x + x + x + x + x + x);
}
```

حال تعداد اعشار را به 5 تا تغییر دهید. چه مشاهده میکنید؟ به عنوان راهنمایی، مقدار عدد 0.3 را در سیستم استاندارد IEEE754 در این لینک بررسی نمایید. علت را برای دستیاران آموزشی توضیح دهید.(قسمت 10)

6- انجام دهید!(امتیازی)

قطعه کد زیر را در یک پروژه جدید اجرا کنید. سپس به عنوان ورودی عدد 115 را وارد کنید.

```
#include <stdio.h>
int main() {
    char x;
    printf("Enter a number:\n");
    scanf("%d", &x);
    printf("%c\n", x);
}
```

مشاهده می‌کنید که برنامه پس از اجرا با خطای runtime مواجه می‌شود. علت این خطا چیست؟

توجه: ممکن است پس از اجرا با خطا مواجه نشوید که این به دلیل رشد و پیشرفت کامپایلرها می‌باشد. ولی در نسخه‌های قدیمی، این خطا رخ خواهد داد.

شاید متوجه شباهت این کد و قطعه کد دوم انجام دهید 3 شده باشد. این دو قطعه کد چه تفاوتی دارند؟ چرا در آنجا با خطا مواجه نمی شویم؟

قسمت 11: نتیجه را با دستیاران آموزشی مطرح کنید.

موفق باشید.

تهیه و تنظیم: امیرمرتضی رضائی