

# შესავალი დაპროგრამებაში

ლექტორი:

სალომე ონიანი

ტელ. 571 39 40 22

ელ. ფოსტა [salome.oniani@iliauni.edu.ge](mailto:salome.oniani@iliauni.edu.ge)

# კურსის ხანგრძლივობა

<b>15 კვირა:</b>	ლექცია - 13 სთ
	პრაქტიკული მეცადინეობა - 25 სთ
	შუალედური გამოცდა 1 - 2სთ
	შუალედური გამოცდა 2 - 2სთ
	საბოლოო შეფასება - 3 სთ

# ძირითადი თემები

- დაპროგრამების ენის ძირითადი კონსტრუქციები
- ფუნქციები
- კომბინირებული ტიპები
- მიმთითებლები
- ფაილები

# შეფასების კომპონენტები:

შეფასების ფორმა	შეფასების კომპონენტი	მინიმალური ზღვარი
შუალედური შეფასება	შუალედური გამოცდა 1 – 20 ქულა შუალედური გამოცდა 2 – 20 ქულა პრაქტიკული მეცადინეობის შეფასება - 30 ქულა	21 ქულა
დასკვნითი შეფასება	დასკვნითი შეფასება - 30 ქულა	12 ქულა

# პრაქტიკული მეცადინეობის შეფასება

პრაქტიკული მეცადინეობის შეფასება ხდება სტუდენტის მიერ პრაქტიკულ მეცადინეობებზე და დამოუკიდებლად შესრულებული დავალებების ჯამური შეფასების შედეგად. კურსის განმავლობაში, დამოუკიდებლად და პრაქტიკულ მეცადინეობებზე, სტუდენტმა უნდა შეასრულოს 10 დავალება, რომლის მიზანია კონკრეტული ამოცანისთვის საჭირო პროგრამის დაწერა დაპროგრამების ენა C-ის გამოყენებით.

15 სასწავლო კვირა

I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV
	3 ქულა	3 ქულა	3 ქულა	3 ქულა	3 ქულა			3 ქულა	3 ქულა	3 ქულა			3 ქულა	3 ქულა

# შუალედური გამოცდების შეფასება

შუალედური გამოცდა ტარდება ორჯერ და თითოეული მოიცავს ორ 10 ქულიან საკითხს.

15 სასწავლო კვირა														
I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	XIII	XIV	XV
						20 ქულა					20 ქულა			

საბოლოო გამოცდა შედგება სამი 10 ქულიანი საკითხისგან.

# ძირითადი ლიტერატურა

„დაპროგრამების ენა C“, ირინა ხუციშვილი, თინა დავითაშვილი, ნათელა არჩვაძე, ამირან ჩიტაღაძე, კობა გელაშვილი. თბილისის უნივერსიტეტის გამომცემლობა, 2009 წელი.

Herbert Schildt, C/C++ Programmer's Reference, Third Edition, MacGraw-Hill Comp., 2005.

C Programming, Greg Perry and Dean Miller, 2014

# სასწავლო ცხრილი

## პარასკევი

ლექცია: 14:00 სთ

აუდიტორია: T304

პრაქტიკული მეცადინეობა: 15:00 სთ

აუდიტორია: T304

16:00 სთ

აუდიტორია: T304

მეცადინეობის ხანგრძლივობა: 50 წთ

შესვენება: 10 წთ



# სასწავლო კვირა I

- დაპროგრამების ენების მიმოხილვა
- მონაცემთა ტიპის ცნება
- რიცხვითი მონაცემები
- მთელი და არამთელი რიცხვები
- მოდიფიკატორები
- სათანადო ტიპის ცვლადის განსაზღვრა
- ცვლადის ინიციალიზაცია

# პროგრამული ენების 5 ძირითადი ტიპი

- პროცედურული პროგრამირების ენები

ეს არის კომპიუტერული პროგრამირების ენა, რომელიც იყენებს პროგრამული ბრძანებების თანმიმდევრობას. ესეთ ენებს წარმოადგენს BASIC, C, FORTRAN, Java და Pascal.

# პროგრამული ენების 5 ძირითადი ტიპი

- პროცედურული პროგრამირების ენები
- ფუნქციონალური პროგრამირების ენები

ფუნქციონალური პროგრამირების ენები სპეციალურად შექმნილია სიმბოლური გამოთვლებისა და სიების დამუშავებისათვის. ფუნქციონალური პროგრამირებას საფუძვლად უდევს მათემატიკური ფუნქციები. ესეთი პროგრამირების ენებია: Lisp, Python, Erlang, Haskell და Clojure.

# პროგრამული ენების 5 ძირითადი ტიპი

- პროცედურული პროგრამირების ენები
- ფუნქციონალური პროგრამირების ენები
- ობიექტზე ორიენტირებული პროგრამირების ენები

OOP - არის მაღალი დონის პროგრამირების ენა, რომელიც დაფუძნებულია ობიექტზე ორიენტირებულ მოდელზე. ანუ პროგრამირების დროს იქმნება ობიექტი, რომელსაც აქვს გარკვეული ფუნქციები. ობიექტზე ორიენტირებული პროგრამირების ენების მაგალითებია Java, C++ და Smalltalk.

# პროგრამული ენების 5 ძირითადი ტიპი

- პროცედურული პროგრამირების ენები
- ფუნქციონალური პროგრამირების ენები
- ობიექტზე ორიენტირებული პროგრამირების ენები
- სკრიპტული პროგრამირების ენები

სკრიპტული პროგრამირების ენები ინტერპრეტირდებიან, ანუ ის კოდის გაშვებისას ითარგმნება მანქანურ კოდად და არ საჭიროებს წინაწარ კომპირებას. ესეთი პროგრამირების ენებია JavaScript, Python და Ruby.

# პროგრამული ენების 5 ძირითადი ტიპი

- პროცედურული პროგრამირების ენები
- ფუნქციონალური პროგრამირების ენები
- ობიექტზე ორიენტირებული პროგრამირების ენები
- სკრიპტული პროგრამირების ენები
- ლოგიკური პროგრამირების ენები

ლოგიკური პროგრამირება არის კომპიუტერული პროგრამირების პარადიგმა, სადაც პროგრამის კოდში გამოხატულია ფაქტები და წესები, რომელბიც სისტემაში არსებობს.

# რა არის პროგრამირება

მოდი შევთანხმდეთ რომ კომპიუტერი არაა ჭკვიანი, ადამიანები ქმნიან მას ჭკვიანად. ასე რომ თქვენ შეგიძლიათ უთხრათ კომპიუტერს თუ რა გააკეთოს, შესაბამისად ის ემორჩილება თქვენს მითითებებს. შესაბამისად, ის მისი მუშაობის პროცესში ამუშავებს იმ მონაცემებს, რომელთაც თქვენ აწვდით

## პროგრამისტი

პროგრამისტი არის პიროვნება, რომელიც ეუბნება კომპიუტერს თუ რა უნდა გააკეთოს და აძლევს მას ძალიან დეტალურ ინსტრუქციებს. ინსტრუქციის გარეშე, კომპიუტერი უსარგებლოა, ამრიგად კომპიუტერის შეცდომა პროგრამისტის შეცდომაა ☺



## პროგრამა

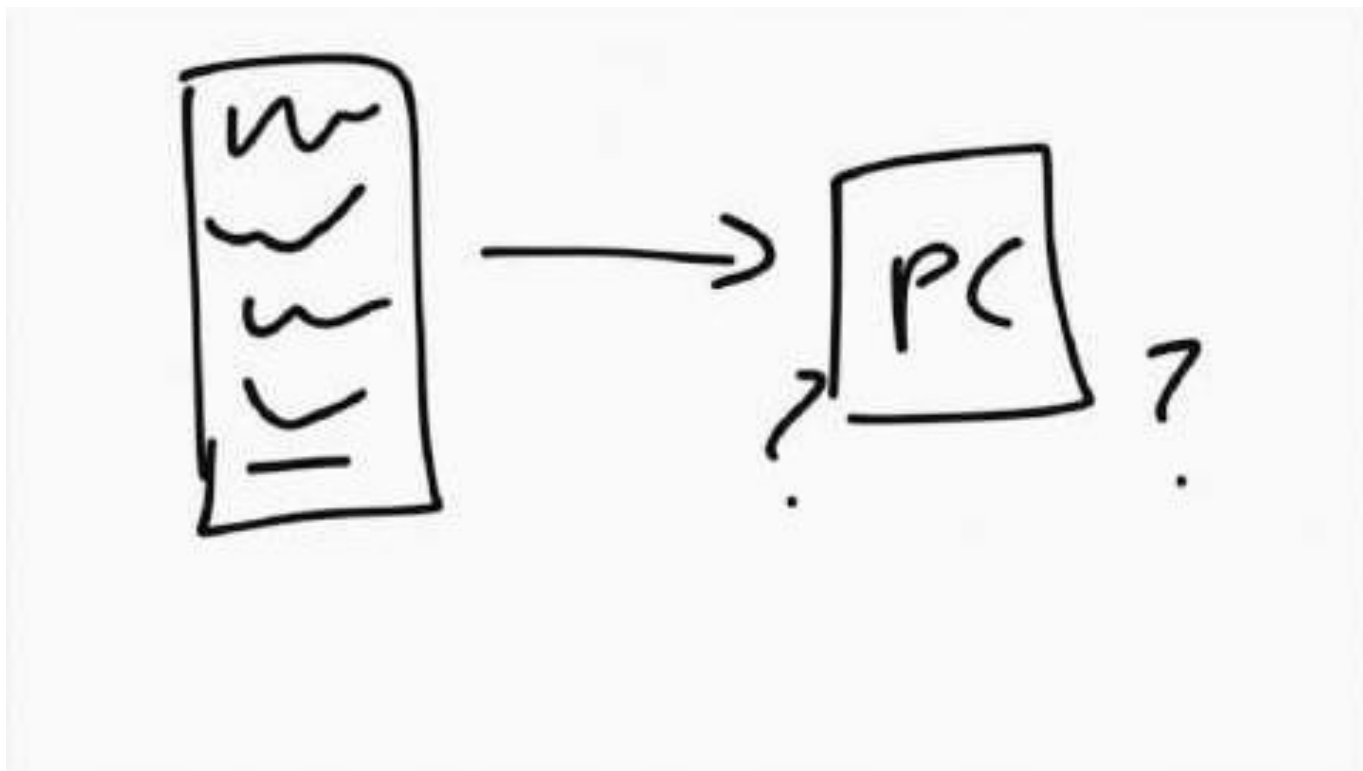
დეტალიზირებულ ინსტრუქციას, რომელსაც თქვენ მიაწოდებთ კომპიუტერს, რათა შეასრულოს თქვენი კონკრეტული დავალება, პროგრამისტები ეძახიან პროგრამას.

პროგრამას აქვს შედეგები **output**, ხოლო როდესაც მას უშვებ შესრულებაზე პროგრამისტები ეძახიან **run ან execute**.

# რა გჭირდება C-ზე პროგრამირებისას?



იმისათვის რომ თქვენს კომპიუტერში C პროგრამის ჩაწერა და შესრულებაზე გაშვება შეძლოთ, გჭირდებათ C-ს კომპილატორი. კომპილატორი ახორციელებს თქვენს მიერ დაწერილი კოდის builds ან გადაყვანას კომპიუტერისათვის გასაგებ ენაზე და ამზადებს თქვენს პროგრამულ კოდს შესარულებაზე გასაშვებათ, რის შემდეგაც თქვენ ხედავთ პროგრამული კოდის შესრულების შედეგს.



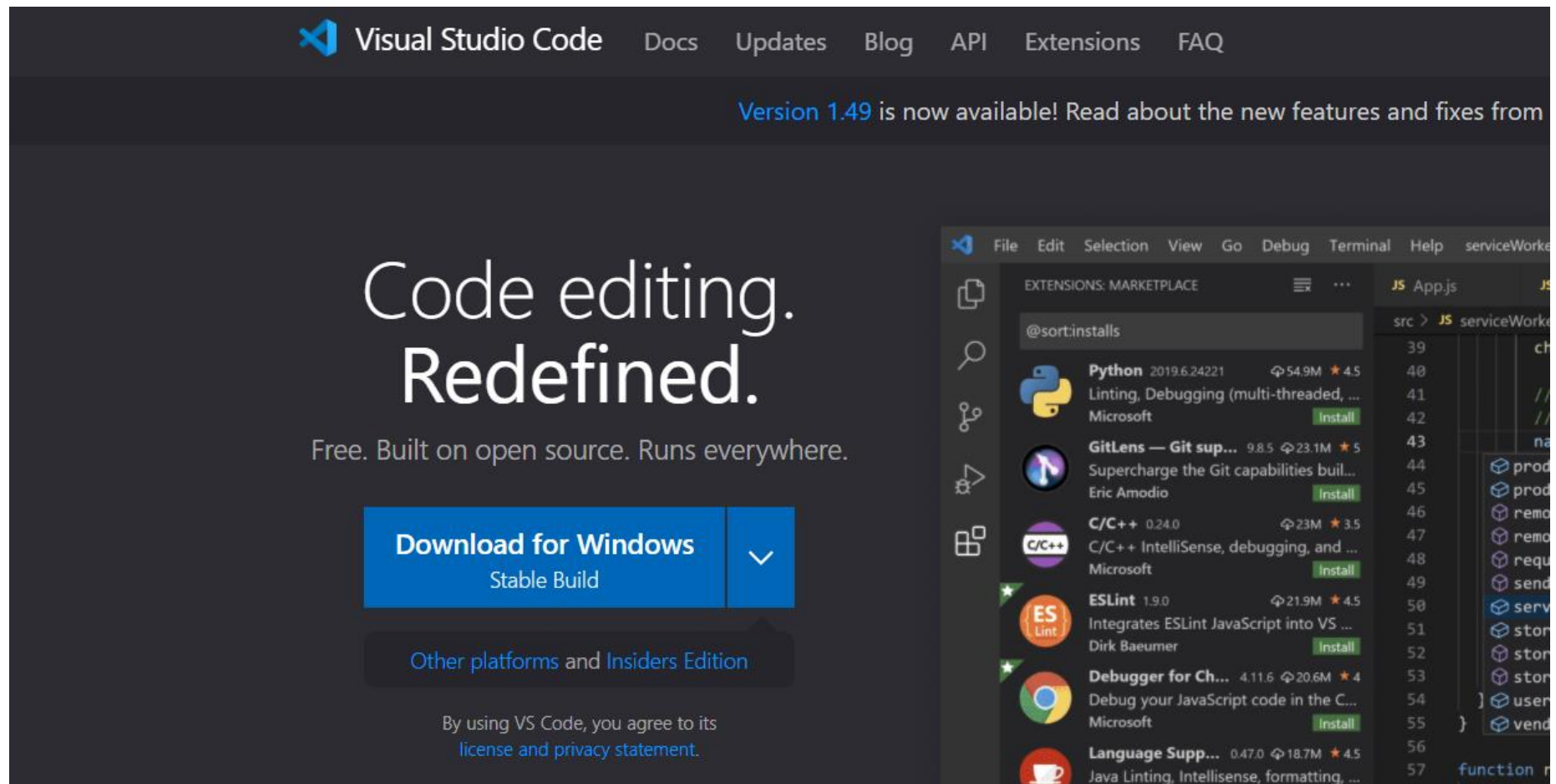
როგორ მუშაობს კომპილტორი?

<https://www.youtube.com/watch?v=8zmaYXNiwZk>

# Visual Studio Code ინსტალაცია

Visual Studio Code შეგიძლიათ გადმოიწეროთ ლინკიდან:

<https://code.visualstudio.com/>





<https://www.youtube.com/watch?v=Ubfgi4NoTPk>

# პროგრამირების პროცესი

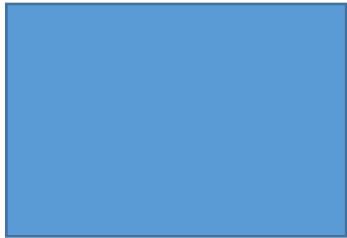
1. იდეის ჩამოყალიბება
2. რედაქტორის გამოყენება

.c გაფართოების ფაილი

3. პროგრამის კომპილაცია
4. პროგრამული შეცდომების შესწორება
5. პროგრამის გაშვება

# მეხსიერებაში ცვლადისათვის ადგილის გამოყოფა

კომპიუტერის მეხსიერებაში გამოყოფილი ადგილი, რომელიც  
განკუთვნილია გარკვეული ტიპის მონაცემების შესანახად.



# მონაცემთა ტიპი

1. სიმბოლოები
2. Boolean
3. მთელი რიცხვები
4. რეალური რიცხვები (ცოცავ მძიმეიანი რიცხვები)



# სიმბოლური მონაცემების ტიპი

სიმბოლური ტიპი არის ნებისმიერი სიმბოლო, რომლის გამოსახვა შეუძლია თქვენს კომპიუტერს, სულ არსებობს 256 სიმბოლო. თითოეული მათგანის რიცხვითი კოდი შეგიძლიათ იპოვნოთ ასკი კოდების ცხრილში.

ასე რომ შეგიძლიათ ნებისმიერი სიმბოლო გამოიყენოთ თქვენს პროგრამულ კოდში. სიმბოლური ტიპის მონაცემები პროგრამულ კოდში აუცილებლად ისმევა აპოსტროფებში, რისი საშუალებითაც კომპილატორი ხვდება რომ სიმბოლურ მონაცემებთან უწევს მუშაობა.

**არსებობს 256 სიმბოლო**

**ASCII - askee → გვერდი 265**

A a 4 % Q ! + = ]

კოდში

'A' 'a' '4' '%' '!' '+' '=' ']'

# სტრიქონული მონაცემების ტიპი

**string**      **"C Programming"**

მაგრამ თუ პროგრამულ კოდში გჭირდება ერთზე მეტი სიმბოლოს გამოყენება უნდა გამოიყენოთ ორმაგი აპოსტროფი და უკვე იმუშავებთ სტრიქონებთან. String-ის ტიპის ცვლადებს უფრო ღრმად შემდეგ ლექციებზე გავეცნობით.

# რიცხვითი მონაცემების ტიპი

**integer**

.10    54    0    -121    -68    752

მათემატიკაში მას უწოდებენ მთელ რიცხვთა სიმრავლეს და მოიცავს დადებით და უარყოფით მთელ რიცხვებს.

# რიცხვითი მონაცემების ტიპი

**integer**

.10    54    0    -121    -68    752

როგორ ფიქრობთ შესაძლებელია **integer** ტიპი მონაცემი დაიწყოს 0-ით?

**მაგალითად 032**

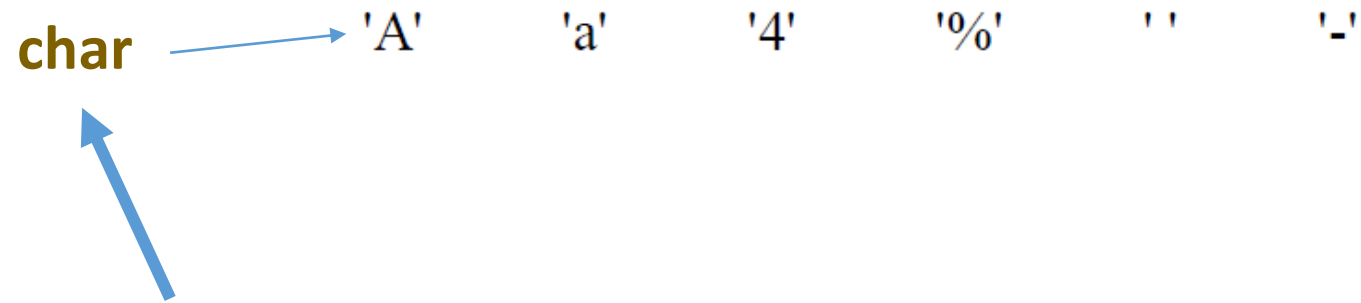
# რიცხვითი მონაცემების ტიპი

რეალური რიცხვები (ცოცავ მძიმიანი რიცხვები)

.  
547.43      0.0      0.44384      9.1923      -168.470

# მონაცემთა ცვლადის მოდიფიკატორები

**char** → 'A'    'a'    '4'    '%'    ' '    '␣'



სიმბოლური მონაცემთა ტიპის მოდიფიკატორი

# მონაცემთა ცვლადის მოდიფიკატორები

**int** → 10    54    0    -121    -68    752




მთელი მონაცემთა ტიპის მოდიფიკატორი

**დიაპაზონი [-2 147 483 648, 2 147 483 647]**

**დიაპაზონის მოცულობა დამოკიდებულია კომპილატორზე**

# მონაცემთა ცვლადის მოდიფიკატორები

**char**  'A'    'a'    '4'    '%'    ' '    '␣'

**int**  10    54    0    -121    -68    752

**დიაპაზონი [-2 147 483 648, 2 147 483 647]**

**როგორ ფიქრობთ რატომ 2 147 483 647 და რატომ არა 2 147 483 648?**



# მონაცემთა ცვლადის მოდიფიკატორები

**char** → 'A'    'a'    '4'    '%'    ' '    '␣'

**int** → 10    54    0    -121    -68    752

**float** → 547.43    0.0    0.44384    9.1923    -168.470

**double** → 547.43    0.0    0.44384    9.1923    -168.470

**Double და float - არის რეალური ტიპის რიცხვების მოდიფიკატორი**

# ცვლადის ინიციალიზაცია

## ცვლადის ტიპი

ცვლადის სახელი  myData pay94 age\_limit amount QtlyIncome

**მაქსიმუმ ცვლადის სახელში შეგიძლიათ გამოიყენოთ 31 სიმბოლო**

**შეუძლებელია ერთიდაიგივე სახელი გამოყენებული იყოს ორი ცვლადისათვის**

**შეუძლებელია**  94Pay my Age lastname,firstname

ცვლადის ინიციალიზაციისათვის გვჭირდება ცვლადის ტიპი და ცვლადის სახელი. ცვლადის ტიპი განსაზღვრავს თუ რა ტიპის მონაცემებთან უნდა ვიმუშაოთ. ხოლო ცვლადის სახელი კი განსაზღვრავს მესიერების იმ არის სახელს, რომელშიც ინახავთ ცვლადის მნიშვნელობებს. თქვენ შეგიძლიათ გამოიყენოთ ცვლადის სახელად ნებისმიერი სახელი, რომელიც თქვენ გსურთ. ცვლადის სახელში შეგიძლიათ გამოიყენოთ 1-დან 31 -მდე სიმბოლო თუმცა სახვადასხვა კომპილატორებს სხვადასხვა დიაპაზონი აქვთ. დაუშვებელია ცვლადის სახელი იწყებოდეს რიცხვით ან მოიცავდეს სასვენ ნიშნებს.

# ცვლადის ინიციალიზაცია

## მაგალითი

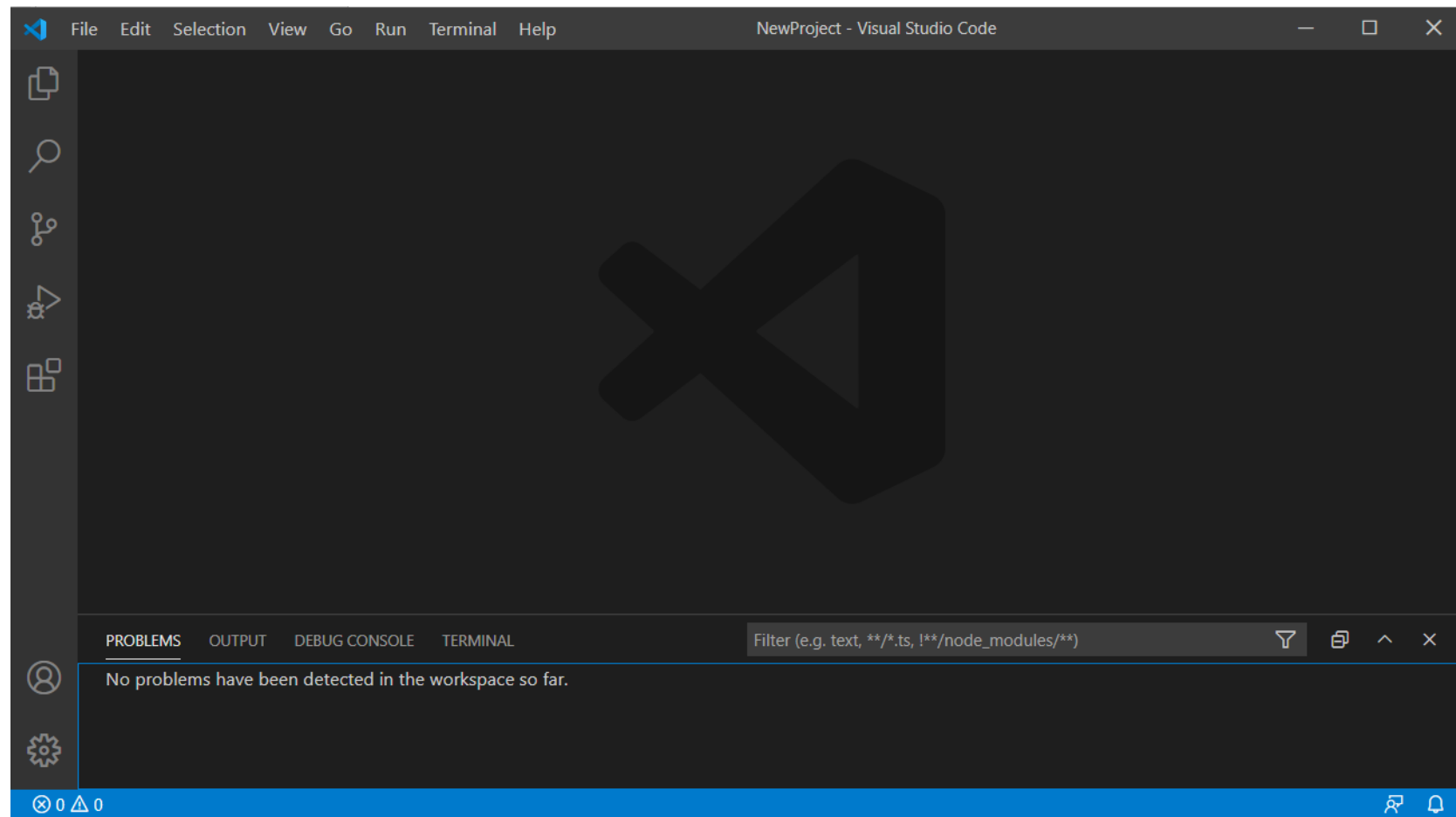
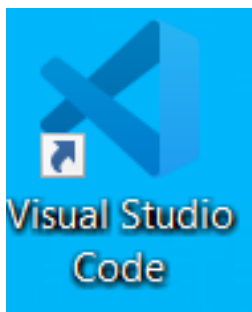
```
char answer;  
int quantity;  
float price;
```

## ასევე შესაძლებელია

```
char first_initial;  
char middle_initial;
```

# პირველი პროგრამული კოდი

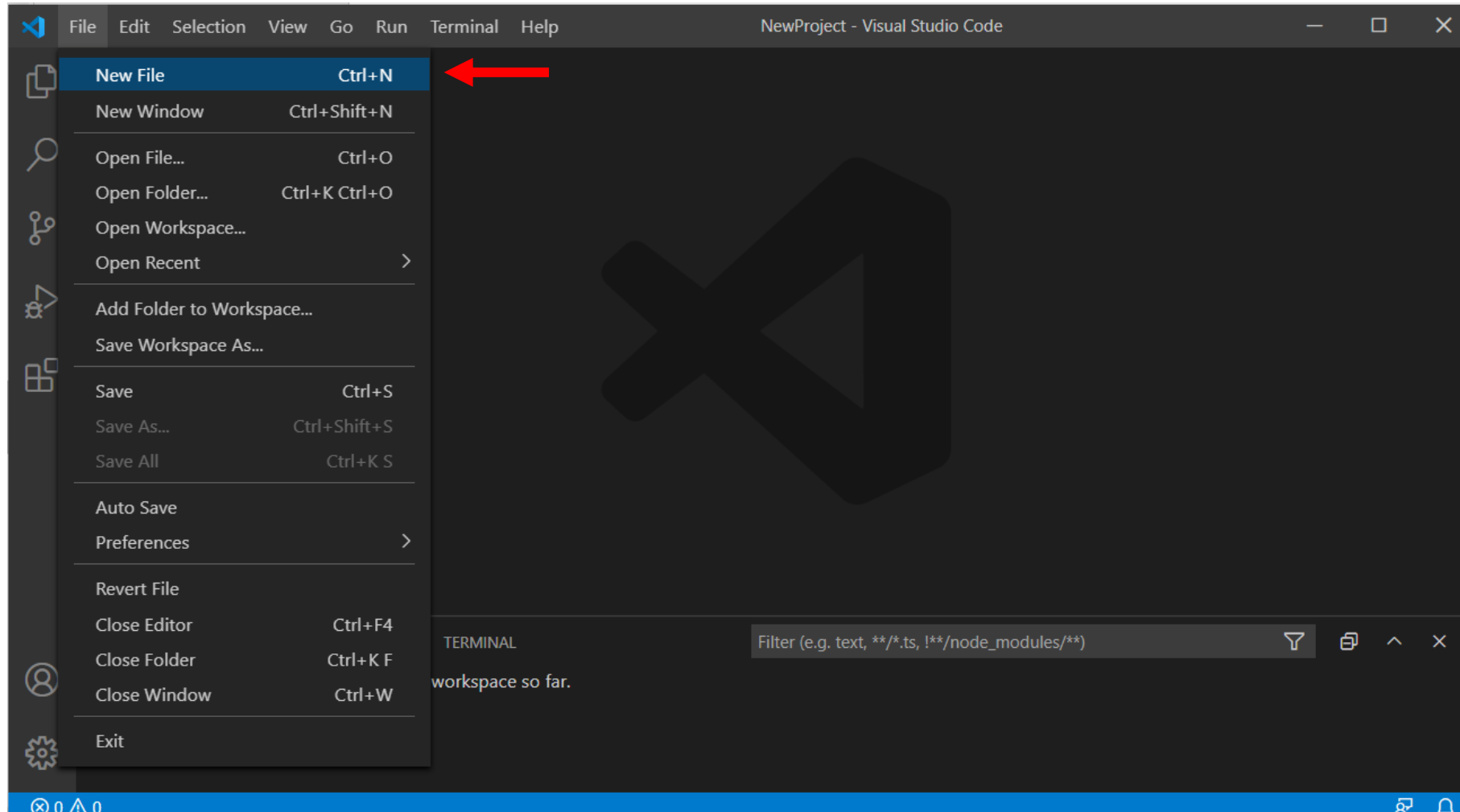
გახდენით Visual  
Studio Code



[https://www.youtube.com/watch?v=77v-Poud\\_io](https://www.youtube.com/watch?v=77v-Poud_io)

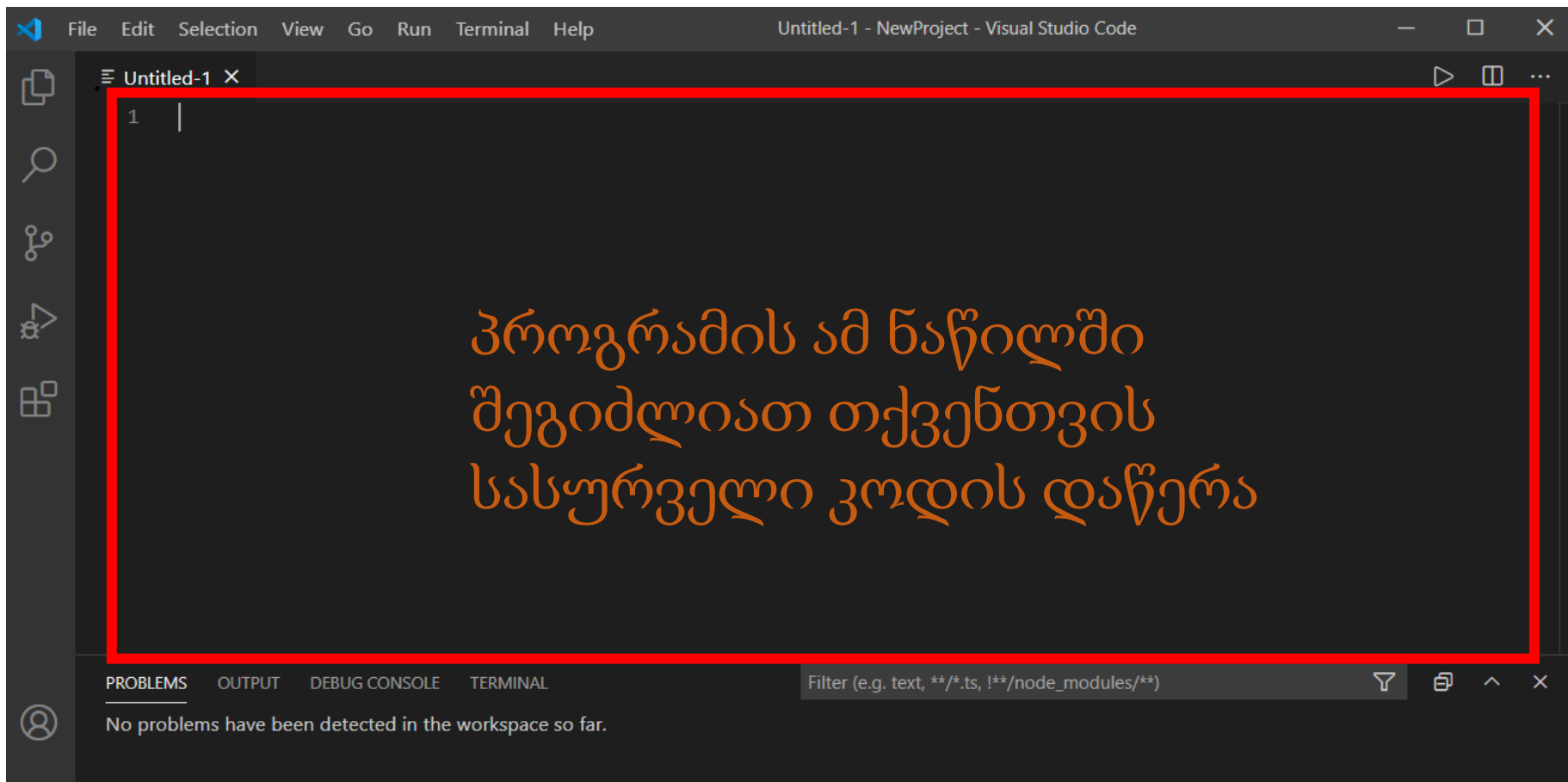
# პირველი პროგრამული კოდი

აირჩიეთ „File“ -> „New File“



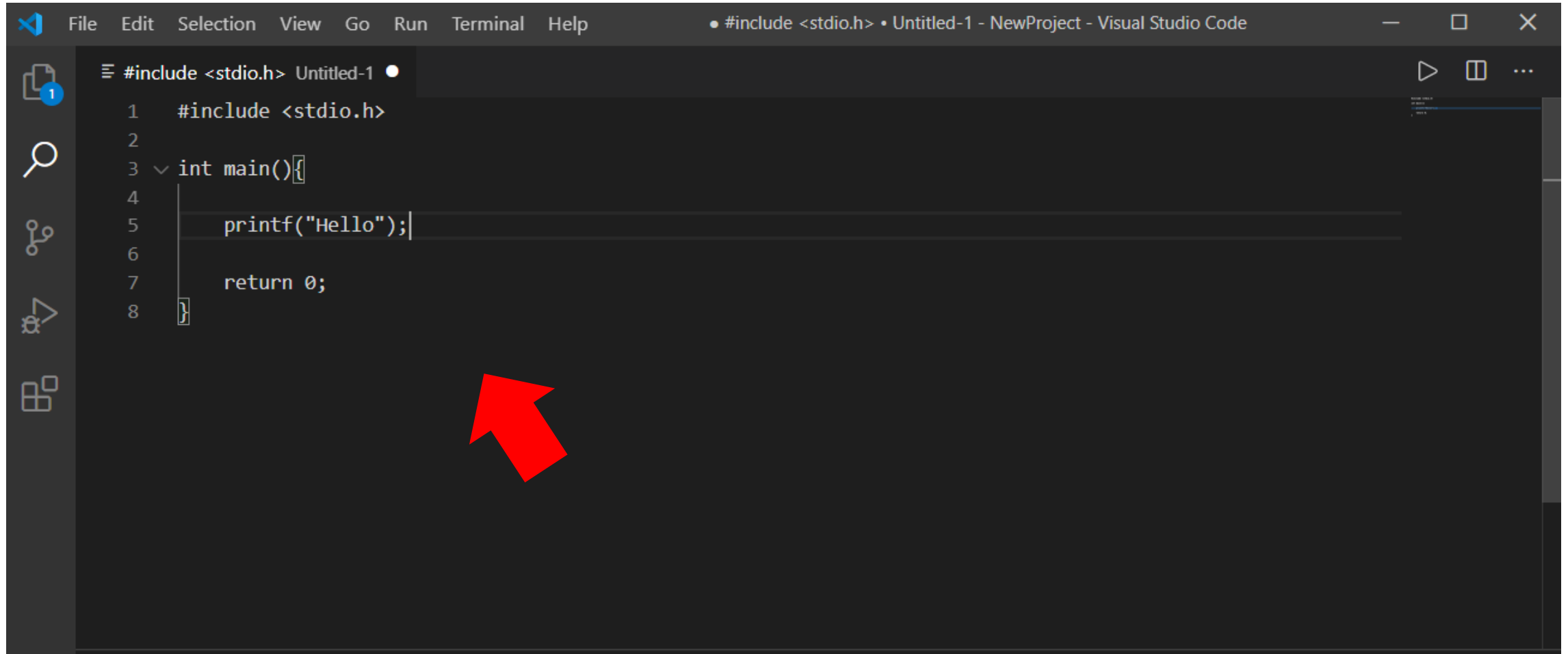
# პირველი პროგრამული კოდი

მიიღებთ შემდეგ შედეგს



# პირველი პროგრამული კოდი

მაგალითად



The image shows a screenshot of the Visual Studio Code editor interface. The title bar at the top reads "File Edit Selection View Go Run Terminal Help" and "• #include <stdio.h> • Untitled-1 - NewProject - Visual Studio Code". The editor window displays a C program with the following code:

```
#include <stdio.h>

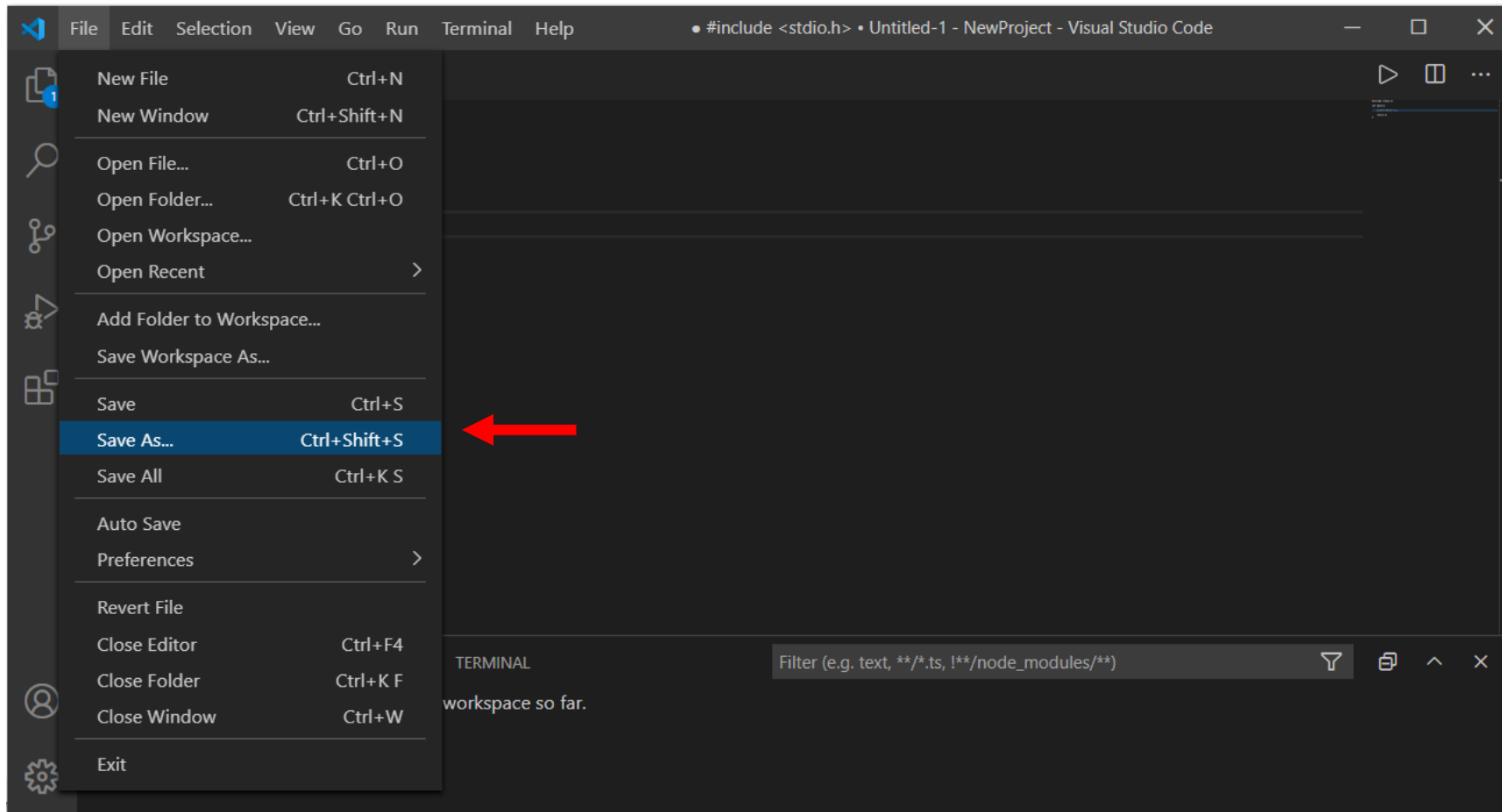
int main(){
    printf("Hello");
    return 0;
}
```

A red arrow points to the code area. The left sidebar contains icons for Explorer, Search, Source Control, Run and Debug, and Extensions. The Explorer icon has a blue circle with the number 1 next to it.

# პირველი პროგრამული კოდი

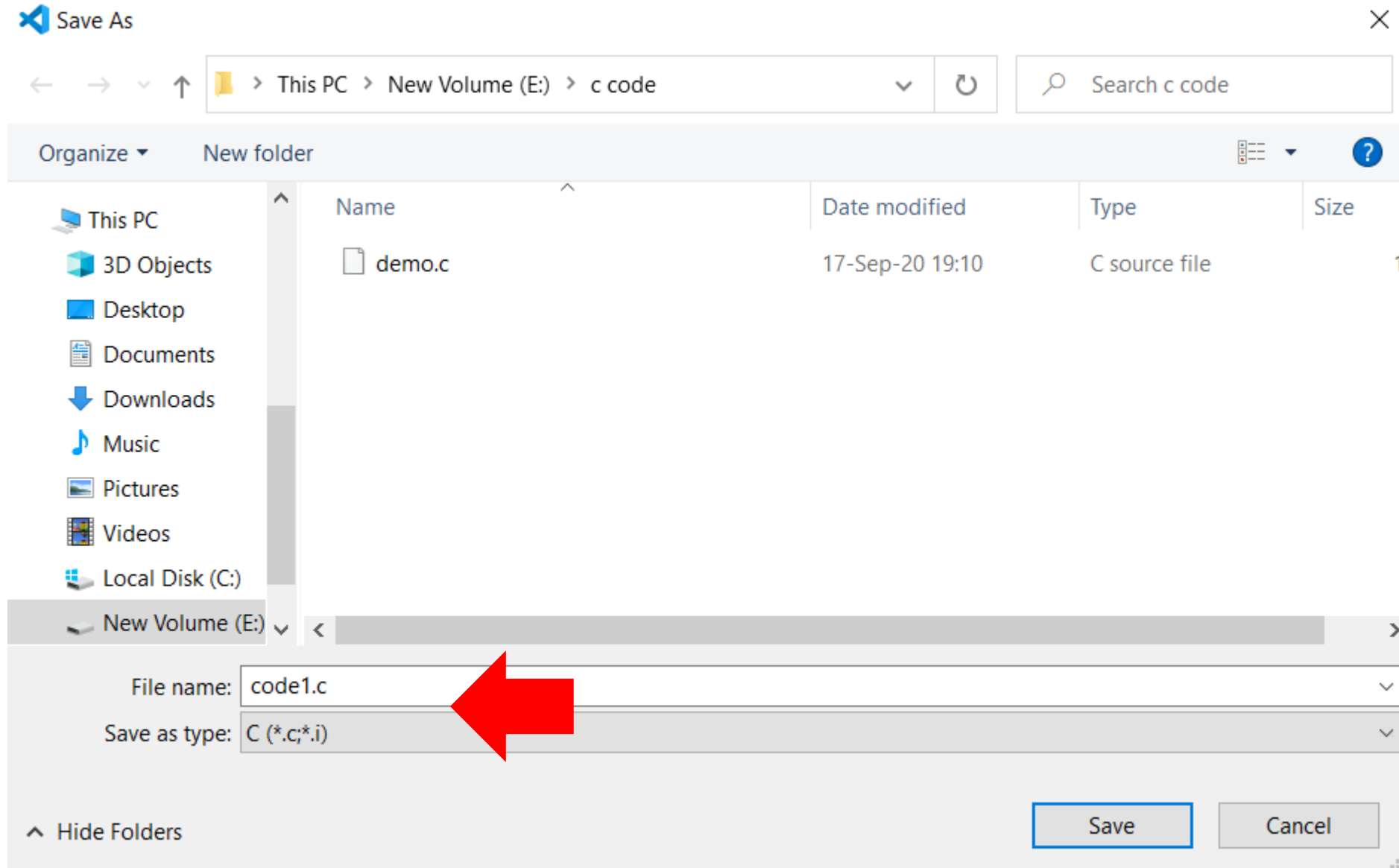
შეინახეთ პროგრამული ფაილი .c გაფართოებით

აირჩიეთ „File“ -> „Save As...“

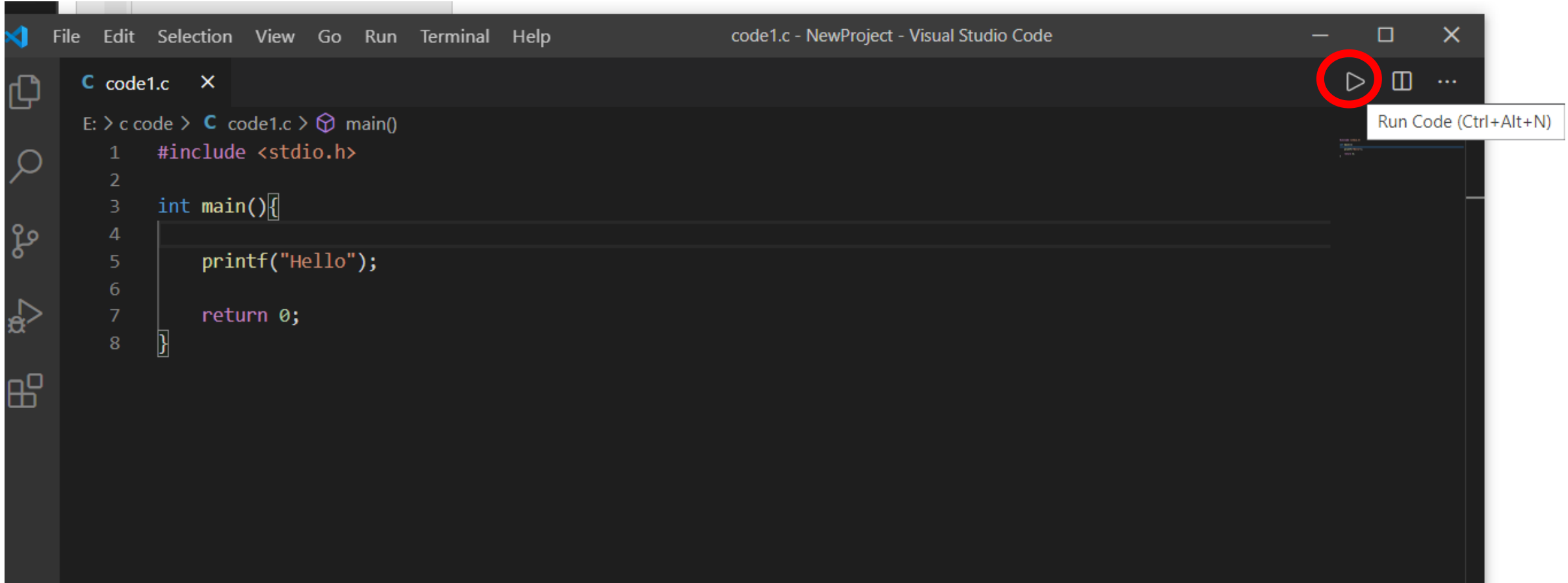




# პირველი პროგრამული კოდი



# პირველი პროგრამული კოდი

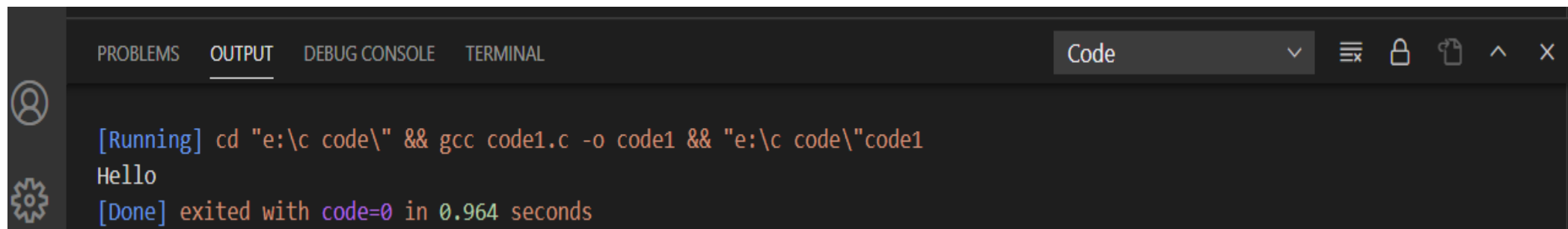


# პირველი პროგრამული კოდი

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

```
int main() {  
    printf("Hello!");  
    return 0;  
}
```

პროგრამის გაშვების შედეგად  
კონსოლის არეში „Hello!“



The screenshot shows a terminal window with a dark background. At the top, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL'. The 'OUTPUT' tab is selected. On the right side of the terminal, there is a 'Code' button and several icons for window management. The terminal output shows the command being executed: `[Running] cd "e:\c code\" && gcc code1.c -o code1 && "e:\c code\"code1`. Below the command, the output 'Hello' is displayed. At the bottom, it says `[Done] exited with code=0 in 0.964 seconds`. On the left side of the terminal, there are icons for a user profile and a gear (settings).

```
[Running] cd "e:\c code\" && gcc code1.c -o code1 && "e:\c code\"code1  
Hello  
[Done] exited with code=0 in 0.964 seconds
```

# პირველი პროგრამული კოდი

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

```
int main() {  
    printf("Hello!");  
    return 0;  
}
```



იმისათვის რომ C ენაში გამოვიყენოთ სხვადასხვა სახის ბრძანებები ჩვენ გვჭირდება მათი განმსაზღვრელი ბიბლიოთეკები, ან პროგრამული ფაილები, რომლებში ჩაწერილია სხვადასხვა პროგრამული ბრძანების განსაზღვრებები, მათი ფუნქციები, პარამეტრები და ა.შ.

ჩვენ შემთხვევაში თუ გვინდა რომ პროგრამულ კოდში გამოვიყენოთ printf ბრძანება აუცილებლად უნდა გამოვიყენოთ stdio.h ბიბლიოთეკა. ხოლო მთლიანი ჩანაწერი #include <stdio.h> ნიშნავს რომ თქვენს პროგრამულ კოდში შემოგავთ stdio.h ბიბლიოთეკაში განსაზღვრული პროგრამული ბრძანებების გამოყენებისათვის საჭირო მხარდაჭერა.

# პირველი პროგრამული კოდი

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

```
int main(){  
    printf("Hello!");  
    return 0;  
}
```


ეს არის მთავარი ფუნქცია, რომლის შიგნით ვწერთ ჩვენთვის სასურველ ბრძანებებს. მას აუცილებლად თან ახლავს ფიგურული ფრჩხილები, რომელბითაც მივითუთებს მთავარი ფუნქციის დასაწყისს და დასასრულს. ხოლო მთავარ ფუნქციაში განსახორციელებელ მოქმედებებს ვწერთ ფიგურული ფრჩხილების შიგნით.

# პირველი პროგრამული კოდი

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

```
int main() {  
    printf("Hello!");  
    return 0;  
}
```

Main ფუნქცია არის მთელი ტიპის ცვლადი და აუცილებლად მოითხოვს რაიმე მნიშვნელობის დაბრუნებას. ხოლო ფუნქციიდან მნიშვნელობის დასაბრუნებლად ვიყენებთ return ბრძანებას და შემდეგ ვწერ დასაბრუნებელ რიცხვს. სტანდარტულად მთავარ ფუნქციაში აბრუნებენ 0-ს.




# პირველი პროგრამული კოდი

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

```
int main() {  
    printf("Hello!");  
    return 0;  
}
```

და ბოლოს printf() პროგრამული ბრძანებით ჩვენ შეგვიძლია განვახორციელოთ რაიმე მონაცემების კონსოლში ანუ კომპიუტერის შავ ეკრანზე დაბეჭდვა. ჩვენს კონკრეტულ შემთხვევაში ვბეჭდავთ სტრიქონს სახელდობრ Hello-ს



## დაბეჭდეთ თქვენი სახელი და გვარი



```
[Running] cd "e:\c code\" && gcc code1.c -o code1 && "e:\c code\"code1  
სალომე ონიანი  
[Done] exited with code=0 in 0.951 seconds
```



# პროგრამული კოდი

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

```
int main() {  
    printf("Group: 1 \n");  
    printf("Name: Salome Oniani \n");  
    printf("University: Ilia State University");  
    return 0;  
}
```

**\n** - გამოიყენება კურსორის ახალ სტრიქონზე გასადატანად, მისი გამოყენებით შეგვიძლია ჩვენთვის სასურველი ინფორმაცია სტრიქონებად გამოვიტანოთ კონსოლის ეკრანზე

ეკრანზე გამოვითვანოთ შედეგი ინფორმაცია:



- Group: 1  
Name: your name  
University: Ilia State University

## ვიპოვოთ შეცდომა



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

```
printf("Group: 1 \n");
```

```
int main() {  
    printf("Name: Salome Oniani \n");  
    printf("University: Ilia State University");  
    return 0;  
}
```

## ვიპოვოთ შეცდომა



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

```
int main() {  
    printf(Group: 1 \n);  
    printf("Name: Salome Oniani \n");  
    printf("University: Ilia State University");  
    return 0;  
}
```

## ვიპოვოთ შეცდომა



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

```
int main() {
```

```
    printf(2017);  
    return 0;
```

```
}
```

# მთელი რიცხვების კონსოლში დაბეჭდვა

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

```
int main() {
```

```
    printf("Year: %d", 2017);
```

```
    return 0;
```

```
}
```

**%d** - გამოიყენება კონსოლის  
ეკრანზე მთელი ტიპის მონაცემების  
დასაბეჭდად

# ათწილადი რიცხვების კონსოლში დაბეჭდვა

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

```
int main() {
```

```
    printf("Weight: %.1f kg", 8.5);
```

```
    return 0;
```

```
}
```

**%.1f** - გამოიყენება კონსოლის

ეკრანზე რეალური ტიპის  
მონაცემების დასაბეჭდად. სადაც  
წერტილის შემდეგ ვწერთ რიცხვს  
თუ რამდენი ციფრი გვინდა  
გამოვიტანოთ მძინის მერე.

# სიმბოლოების კონსოლში დაბეჭდვა

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

```
int main() {
```

```
    printf("%c", 'A');
```

```
    return 0;
```

```
}
```

**%C** - გამოიყენება კონსოლის  
ეკრანზე სიმბოლური ტიპის  
მონაცემების დასაბეჭდად



## ვიპოვოთ შეცდომა



```
#include <stdio.h>

int main() {
    printf(" %c ", '1')

    return 0;
}
```

## გამოიტანეთ ეკრანზე შემდეგი ინფორმაცია



```
I am learning the C programming language  
I have just completed Chapter 2  
I am 99.9 percent ready to move on to the next chapter!
```

```
-----
```

# მთელი ტიპის ცვლადის აღწერა და მისთვის მნიშვნელობის მინიჭება

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

```
int main() {
```

```
    int year = 2020;  
    printf("Year: %d", year);  
    return 0;
```

```
}
```

- შესარულებაზე გაუშვით პროგრამული კოდი;
- შეცვალეთ `year` ცვლადის მნიშვნელობა 2020 რაიმე სხვა რიცხვითი მნიშვნელობით, მაგალითად `year = 34`, ან `year = 500` ან თქვენთვის სასურველი რიცხვით;
- შემდეგ კვლავ გაუშვით შესარულებაზე პროგრამული კოდი და დააკვირდით კონსოლზე გამოტანილ ინფორმაციას.



აღწერეთ სიმბოლური ტიპის ცვლადი და მიაჩნეთ  
მნიშვნელობა



აღწერეთ რეალური ტიპის ცვლადი და მიაჩნეთ  
მნიშვნელობა

## ვიპოვოთ შეცდომა



```
#include <stdio.h>
int character = a
int main() {
    printf(" %c ", character);
    return 0;
}
```

## ვიპოვოთ შეცდომა



```
#include <stdio.h>
char character = 123;
int main() {
    printf(" %c ", character);
    return 0;
}
```

## ვიპოვოთ შეცდომა



```
#include <stdio.h>
char character = 345;
int main() {
    printf(" %c ", character);
    return 0;
}
```





## 3 ქულიანი დავალება

დავალების შესრულების ვადაა  
08/10/2020 20:00 საათი.

ამოცანა

დაწერეთ პროგრამული კოდი, სადაც აღწერილი იქნება „ტელეფონის ნომერის“, „სიმაღლის“ და „საყვარელი ციფრის“ ამსახველი ცვლადები, ცვლადებში შეინახეთ შესაბამისი მნიშვნელობები და შემდეგ დაბეჭდეთ მათში ჩაწერილი მონაცემები.

შეფასების კრიტერიუმები და დავალების გამოგზავნის ინსტრუქცია იხილეთ შემდეგ გვერდზე



## 3 ქულიანი დავალება

დავალების შესრულების ვადაა  
08/10/2020 20:00 საათი.

### შეფასება

1. შესრულებულ დავალებაში გამოყენებულია ცვლადის დანიშნულების შესაბამისი მონაცემთა ტიპი და ცვლადის სახელი ცვლადის დანიშნულების შესაბამისია - 1 ქულა
2. შესრულებულ დავალებაში ცვლადებში შენახულია ცვლადის ტიპის შესაბამისი მონაცემი - 1 ქულა
3. შესრულებულ დავალებაში მოცემული პროგრამული კოდი ბეჭდავს ცვლადებში შენახულ მონაცემებს და არ არის კომპილაციის შეცდომები - 1 ქულა

შესრულებული დავალებების პროგრამული კოდები დააკოპირეთ ვორდის ფაილში და გამომიგზავნეთ მეილზე: [salome.oniani@iliauni.edu.ge](mailto:salome.oniani@iliauni.edu.ge)

ვორდის ფაილში უნდა იყოს ამოცანის პირობა და პირობის ქვეშ მისი შესაბამისი პროგრამული კოდი.

გმადლობთ ყურადღებისთვის!