

Lecture 4: main() Function

In this section, I'll start with the beginning of every C program, the `main()` function. Let's first, However, talk philosophically about what a function is. From a programming perspective, functions allow you to group a logical series of activities, or program statements, under one name. For example, suppose I want to create a function called `bakeCake`. My algorithm for baking a cake might look like this:

Mix wet ingredients in mixing bowl

Combine dry ingredients

Spoon batter into greased baking pan

Bake cake at 350 degrees for 30 minutes

Anyone reading my code will see my function called `bakeCake` and know right away that I'm trying to bake cakes.

Functions are typically not static, meaning they are living and breathing entities, again philosophically, that take in and pass back information. Thus, my `bakeCake` function would take in a list of ingredients to bake (called parameters) and return back a finished cake (called a value).

Algorithms

An algorithm is a finite step-by-step process for solving a problem. It can be as simple as a recipe to bake a cake, or as complicated as the process to implement an autopilot system for a 747 jumbo jet.

Algorithms generally start off with a problem statement. It is this problem statement that programmers use to formulate the process for solving the problem. Keep in mind that the process of building algorithms and algorithm analysis occurs before any program code has been written.

The `main()` function is like any other programming function in that it groups like activities and can take in parameters (information) and pass back values (again, information). What makes the `main()` function unique from other functions, however, is that the values it returns are returned to the operating system. Other functions that you will use and create in this course return values back to the calling C statement inside the `main()` function.

In this course I will use `main()` functions that are void of parameters (functions that do not take parameters) and do not return values.

`Main()`

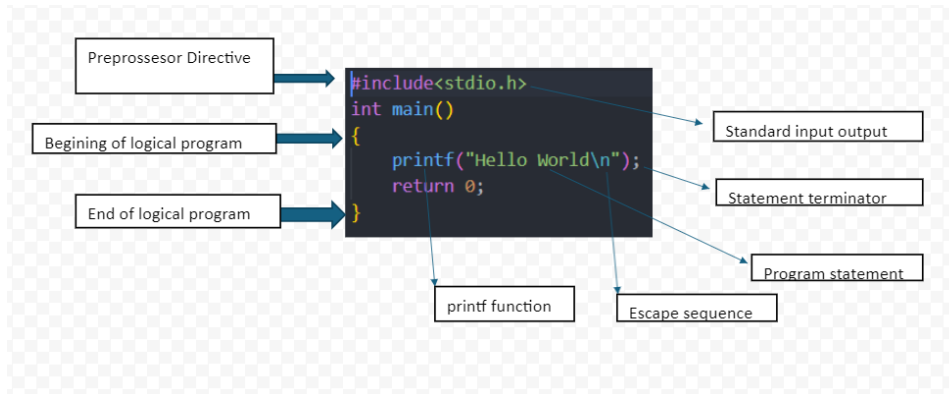
```
{  
  
}
```

As the preceding example shows, the `main()` function begins with the keyword `main` and is followed by two empty parentheses `()`. The parentheses are used to encompass parameters to be passed to the `main()` function.

Coutino: C is a case-sensitive programming language. For example, the function names main(), Main(), and MAIN() are not the same. It takes extra computing resources to NOT be case-sensitive as input devices such as keyboards distinguish between cases.

Following the parentheses are two braces. The first brace denotes the beginning of a logical programming block and the last brace denotes the end of a logical programming block. Each function implementation requires that you use a beginning brace, {, and a closing brace, }.

The following program code demonstrates a complete, simple C program. From this code, you will learn how single program statements come together to form a complete C program.



Lecture 4: main() Function (Hindi)

इस खंड में, मैं हर C प्रोग्राम की शुरुआत, main() फ़ंक्शन के साथ शुरुआत करूंगा। लेकिन पहले, आइए फ़ंक्शन के बारे में दार्शनिक तौर पर बात करते हैं। प्रोग्रामिंग के दृष्टिकोण से, फ़ंक्शन आपको एक नाम के अंतर्गत लॉजिकल श्रृंखला की गतिविधियाँ, या प्रोग्राम स्टेटमेंट्स, समूहित करने की अनुमति देते हैं। उदाहरण के लिए, मान लीजिए मैं एक फ़ंक्शन बनाना चाहता हूँ जिसका नाम bakeCake है। मेरी केक बेक करने की एल्गोरिथम इस प्रकार दिख सकती है:

मिक्सिंग बाउल में गीली सामग्री मिलाएं

सूखी सामग्री को मिलाएं

ग्रीस किए हुए बेकिंग पैन में बैटर डालें

केक को 350 डिग्री पर 30 मिनट के लिए बेक करें

मेरे कोड को पढ़ने वाला कोई भी मेरे फ़ंक्शन को `bakeCake` कहते हुए तुरंत जान जाएगा कि मैं केक बेक करने की कोशिश कर रहा हूँ।

फ़ंक्शन आमतौर पर स्थिर नहीं होते, यानी वे जीवित और श्वास लेने वाले इकाइयाँ होती हैं, फिर से दार्शनिक रूप से, जो जानकारी लेती हैं और वापस जानकारी पास करती हैं। इस प्रकार, मेरा `bakeCake` फ़ंक्शन बेक करने के लिए एक सूची की सामग्री लेगा (जिसे पैरामीटर कहा जाता है) और एक समाप्त केक वापस लौटाएगा (जिसे मान कहा जाता है)।

एल्गोरिथ्म

एक एल्गोरिथ्म एक समस्या को हल करने के लिए एक निश्चित कदम-दर-कदम प्रक्रिया है। यह एक केक बेक करने के नुस्खे जितना सरल हो सकता है, या 747 जंबो जेट के लिए एक ऑटोपायलट सिस्टम को लागू करने की प्रक्रिया जितना जटिल हो सकता है।

एल्गोरिथ्म आमतौर पर एक समस्या विवरण के साथ शुरू होते हैं। यह समस्या विवरण ही है जिसका उपयोग प्रोग्रामर समस्या को हल करने की प्रक्रिया को तैयार करने के लिए करते हैं। ध्यान रखें कि एल्गोरिथ्म बनाने और एल्गोरिथ्म विश्लेषण की प्रक्रिया किसी भी प्रोग्राम कोड लिखे जाने से पहले होती है।

`main()` फ़ंक्शन उसमें समूहित गतिविधियों को और पैरामीटर्स (जानकारी) लेने और मान (फिर से, जानकारी)

पास करने के मामले में अन्य प्रोग्रामिंग फ़ंक्शन की तरह ही है। हालांकि, जो चीज़ `main()` फ़ंक्शन को अन्य फ़ंक्शन से अनोखा बनाती है, वह यह है कि इसके द्वारा लौटाए गए मान ऑपरेटिंग सिस्टम को वापस लौटाए जाते हैं। इस

कोर्स में आपके द्वारा उपयोग किए जाने वाले और बनाए जाने वाले अन्य फ़ंक्शन `main()` फ़ंक्शन के अंदर कॉल करने वाले C स्टेटमेंट को मान वापस करते हैं।

इस कोर्स में मैं ऐसे `main()` फ़ंक्शन का उपयोग करूंगा जो पैरामीटर्स (फ़ंक्शन जो पैरामीटर्स नहीं लेते) से रहित होते हैं और मान वापस नहीं करते।

```
main()  
{  
  
}
```

जैसा कि पूर्ववर्ती उदाहरण दिखाता है, `main()` फ़ंक्शन `main` कीवर्ड के साथ शुरू होता है और इसके बाद दो खाली कोष्ठक `()` होते हैं। कोष्ठकों का उपयोग `main()` फ़ंक्शन में पास किए जाने वाले पैरामीटर्स को समाविष्ट करने के लिए किया जाता है।

Coutino: C एक केस-सेंसिटिव प्रोग्रामिंग भाषा है। उदाहरण के लिए, फ़ंक्शन नाम `main()`, `Main()`, और `MAIN()` समान नहीं हैं। केस-सेंसिटिव न होने के लिए अतिरिक्त कंप्यूटिंग संसाधनों की आवश्यकता होती है क्योंकि इनपुट डिवाइस जैसे कि कीबोर्ड केसों के बीच अंतर करते हैं।

कोष्ठकों के बाद दो ब्रेसेज़ होते हैं। पहला ब्रेस एक लॉजिकल प्रोग्रामिंग ब्लॉक की शुरुआत को दर्शाता है और अंतिम ब्रेस एक लॉजिकल प्रोग्रामिंग ब्लॉक के अंत को दर्शाता है। प्रत्येक फ़ंक्शन कार्यान्वयन के लिए आपको एक शुरुआती ब्रेस, `{`, और एक समाप्ति ब्रेस, `}`, का उपयोग करना आवश्यक है।

निम्नलिखित प्रोग्राम कोड एक पूर्ण, सरल C प्रोग्राम का उदाहरण देता है। इस कोड से, आप सीखेंगे कि कैसे एकल प्रोग्राम स्टेटमेंट्स एक पूर्ण C प्रोग्राम बनाने के लिए एक साथ आते हैं।

