



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Οντοκεντρικός Προγραμματισμός

Ενότητα 5: Η ΓΛΩΣΣΑ C++
Συναρτήσεις - Μεταβλητές

ΔΙΔΑΣΚΟΝΤΕΣ: Ιωάννης Χατζηλυγερούδης, Χρήστος
Μακρής

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Συναρτήσεις / Μεταβλητές

Μαθηματικές Συναρτήσεις

- ... Υλοποίηση απλών μαθηματικών λειτουργιών
 - Απαιτείται η συμπερίληψη του header file `<cmath>`
- Παράδειγμα

```
cout << sqrt( 900.0 );
```

- Το αποτέλεσμα είναι η εκτύπωση της τιμής 30.0
- Όλες οι συναρτήσεις της math library επιστρέφουν **double**



Μαθηματικές Συναρτήσεις

Μέθοδος	Περιγραφή
<code>ceil(x)</code>	Στρογγυλοποίηση προς το πάνω (πλησιέστερο ακέραιο)
<code>cos(x)</code>	Συνημίτονο (x σε radians)
<code>exp(x)</code>	Exponential e^x
<code>fabs(x)</code>	Απόλυτη τιμή
<code>floor(x)</code>	Στρογγυλοποίηση προς τα κάτω (πλησιέστερος ακέραιος)
<code>fmod(x, y)</code>	Υπόλοιπο του x/y ως κινητής υποδιαστολής αριθμός
<code>log(x)</code>	Φυσικός Λογάριθμος του x (βάση το e)
<code>log10(x)</code>	Λογάριθμος του x (βάση το 10)
<code>pow(x, y)</code>	Ύψωση του x στην y (x^y)
<code>sin(x)</code>	Ημίτονο (x σε radians)
<code>sqrt(x)</code>	Τετραγωνική ρίζα
<code>tan(x)</code>	Εφαπτομένη (x σε radians)



Ορισμοί Συναρτήσεων

- Πρωτότυπο συνάρτησης
 - Ενημερώνει το μεταγλωττιστή για τον τύπο των παραμέτρων και τον επιστρεφόμενο τύπο της συνάρτησης
 - **int square(int);**
 - Συνάρτηση που λαμβάνει **int** και επιστρέφει **int**
- Κλήση συνάρτησης
 - **square (x) ;**
 - Παρενθέσεις, τελεστής κλήσης συνάρτησης
 - Πέρασμα παραμέτρου x
 - Η συνάρτηση δέχεται το δικό της αντίγραφο των παραμέτρων
 - Αφού τερματίσει περνάει πίσω το αποτέλεσμα



Πρωτότυπα συναρτήσεων

- Αρχικές δηλώσεις των συναρτήσεων ώστε να μπορούν χρησιμοποιηθούν από πρόγραμμα
- Το πρωτότυπο πρέπει να ταιριάζει με τον ορισμό της συνάρτησης

- Function prototype

```
double maximum( double, double, double );
```

- Definition

```
double maximum( double x, double y, double z ){  
    ...  
}
```

- Υπογραφή συνάρτησης

- Το μέρος του πρωτοτύπου με όνομα και παραμέτρους

- `double maximum(double, double, double);`



Header Files (Αρχεία επικεφαλίδων)

- Περιέχουν
 - Πρωτότυπα συναρτήσεων
 - Ορισμούς τύπων και σταθερών
- Τα αρχεία επικεφαλίδων έχουν κατάληξη .h
 - Programmer-defined header files

```
#include "myheader.h"
```
- Library header files

```
#include <cmath>
```



Γεννήτρια τυχαίων αριθμών

- **rand** function (`<cstdlib>`)
 - `i = rand();`
 - Παράγει έναν unsigned integer μεταξύ 0 και RAND_MAX (συνήθως 32767)
- Scaling and shifting
 - Παράδειγμα
 - `i = rand() % 6 + 1;`
 - “`Rand() % 6`” παράγει έναν αριθμό μεταξύ 0 και 5 (scaling)
 - “`+ 1`” δίνει το διάστημα 1 έως 6 (shift)



Παράδειγμα

```
for ( int counter = 1; counter <= 20; counter++ )  
    cout << setw( 10 ) << ( 1 + rand() % 6 );
```

Παραγωγή και εκτύπωση 20 τυχαίων αριθμών από 1 έως 6.



Γεννήτρια τυχαίων αριθμών

- Η χρήση της `rand()` σε επαναλαμβανόμενες εκτελέσεις του προγράμματος
 - Αποδίδει την ίδια ακολουθία αριθμών
- Για να πάρουμε διαφορετικές ακολουθίες αριθμών
 - Χρησιμοποιούμε μια τιμή φύτρο (`seed value`)
 - Αντιστοιχεί σε τυχαίο σημείο εκκίνησης της ακολουθίας
 - Το ίδιο `seed` θα αποδώσει την ίδια ακολουθία
 - **`srand (seed) ;`**
 - Βιβλιοθήκη: `<cstdlib>`
 - Χρησιμοποιείται πριν από τη `rand ()` για να προσδιορίσει το τυχαίο σημείο εκκίνησης



Γεννήτρια τυχαίων αριθμών

- Μπορούμε να χρησιμοποιήσουμε την τρέχουσα ώρα ως φυτό (seed)
 - `srand(time(0));`
 - `time(0);`
 - `<ctime>`
 - Επιστρέφει την τρέχουσα ώρα σε δευτερόλεπτα



Χαρακτηριστικά Μεταβλητών

- Οι μεταβλητές χαρακτηρίζονται από διάφορες ιδιότητες
 - όνομα, τύπο δεδομένων, μέγεθος, τιμή
 - Τάξη αποθήκευσης
 - Για ποιο διάστημα η μεταβλητή υπάρχει στη μνήμη
 - Εμβέλεια (Scope)
 - Το τμήμα του προγράμματος που μια αναφορά στη μεταβλητή είναι έγκυρη
 - Συνδεσιμότητα (Linkage)
 - Όταν ένα πρόγραμμα υλοποιείται σε πολλά αρχεία, ποιά αρχεία μπορούν να τη χρησιμοποιήσουν



Τάξη αποθήκευσης

- **Automatic**

Η μεταβλητή δημιουργείται όταν ο έλεγχος εισέρχεται σ' ένα block εντολών και καταστρέφεται όταν ο έλεγχος εξέρχεται από το block εντολών

- **Static**

Οι μεταβλητές εξακολουθούν να υπάρχουν καθ' όλη τη διάρκεια ζωής του προγράμματος

- **Extern**

Η κατάσταση της μεταβλητής είναι γνωστή σε κάθε συνάρτηση που ακολουθεί



Κανόνες Εμβέλειας

- **Εμβέλεια (Scope)**
 - Η περιοχή του προγράμματος που ένα όνομα (π.χ. μεταβλητής) μπορεί να χρησιμοποιηθεί
- **Εμβέλεια αρχείου (File scope)**
 - Ορίζεται εκτός μιας συνάρτησης, διαθεσιμότητα για όλες τις συναρτήσεις
 - Σφαιρικές μεταβλητές, οι ορισμοί και τα πρωτότυπα συναρτήσεων
- **Εμβέλεια συνάρτησης (Function scope)**
 - Η περιοχή πρόσβασης περιορίζεται μόνο στη συνάρτηση που ορίζεται το όνομα



Κανόνες Εμβέλειας

- **Εμβέλεια μπλοκ (Block scope)**
 - Ξεκινά από το σημείο δήλωσης του ονόματος και τερματίζει στο κλείσιμο του μπλοκ που υποδηλώνεται με το σύμβολο }
 - Τοπικές μεταβλητές, παράμετροι συναρτήσεων
 - οι **static** μεταβλητές επίσης έχουν εμβέλεια μπλοκ
 - Άλλο το Storage class και άλλο η εμβέλεια
- **Εμβέλεια πρωτότυπου συνάρτησης**
 - Λίστα παραμέτρων
 - Τα ονόματα στο πρωτότυπο είναι προαιρετικά
 - Ο μεταγλωττιστής τα αγνοεί



Κανόνες Εμβέλειας

```
#include <iostream>
using std::cout;
using std::endl;

int x = 1;

int main () {
    cout << "global x is " << x << endl;
    int x = 3;
    cout << "local x in main is " << x << endl;
    cout << "global x is " << ::x << endl;
    {
        int x = 5;
        cout << "this block's x is" << x << endl;
    }
    return 0;
}
```

Καθολική εμβέλεια

Τοπική εμβέλεια μέσα στη συνάρτηση

Εμβέλεια κατά το χρόνο ζωής του μπλόκ.

Ο τελεστής :: μας επιτρέπει να πάρουμε την global μεταβλητή x αντί την τοπική με το ίδιο όνομα

```
global x is 1
local x in main is 3
this block's x is 5
```



Κανόνες Εμβέλειας

```
int useStaticLocal( ){  
    static int x = 5;    //  
    x+=5;  
}  
  
int main ( ){  
    for (int i=1; i<=5; i++)  
        cout << useStaticLocal() <<" ";  
}
```

Στατική τοπική μεταβλητή.
Αρχικοποιείται μόνο μια φορά και
διατηρεί την τιμή της μεταξύ των
κλήσεων της συνάρτησης

10 15 20 25 30



Αναδρομή

- Αναδρομικές συναρτήσεις
 - Συναρτήσεις που καλούν τον εαυτό τους
 - Λύνουν μόνον μια βασική περίπτωση
- Παράδειγμα - παραγοντικό

```
int factorial( int number ){  
    if ( number <= 1 )  
        return 1;  
    else  
        return number * factorial( number - 1 );  
}
```

```
int main () {  
    cout << factorial(5) <<" ";  
}
```



Inline Συναρτήσεις

- Inline functions
 - Χρειάζεται το Keyword **inline** πριν από τον ορισμό της συνάρτησης
 - Ζητά από τον μεταγλωττιστή να αντικαταστήσει κάθε κλήση της συνάρτησης στο πρόγραμμα μ' ένα αντίγραφο του κώδικα της συνάρτησης
 - Μειώνει την επιβάρυνση από την κλήση μιας συνάρτησης
 - Ο μεταγλωττιστής μπορεί να αγνοήσει την οδηγία **inline**
 - Καλή περίπτωση για μικρές και συχνά καλούμενες συναρτήσεις



Αναφορές Μεταβλητών

- Οι αναφορές μεταβλητών είναι συνώνυμα άλλων μεταβλητών
 - Ουσιαστικά πρόκειται για την ίδια μεταβλητή
 - Μπορεί να χρησιμοποιηθεί και εντός μιας συνάρτησης

```
int count = 1; // declare integer variable count
int &cRef = count; // create cRef as an alias for count
++cRef; // increment count (using its alias)
```

- Οι αναφορές πρέπει να αρχικοποιούνται όταν δηλώνονται
 - Διαφορετικά, ο μεταγλωττιστής παράγει error
 - Dangling reference
 - Αναφορά σε μη ορισμένη μεταβλητή



Υπερφόρτωση Συναρτήσεων (Function Overloading)

- Υπερφόρτωση Συναρτήσεων
 - Συναρτήσεις με το ίδιο όνομα και διαφορετικές παραμέτρους
 - Υλοποιούν παρόμοια δουλειά
 - Π.χ. συνάρτηση `square` για `ints` και συνάρτηση `square` για `floats`

```
int square( int x) {return x * x;}  
float square(float x) { return x * x; }
```
- Οι συναρτήσεις που είναι υπερφορτωμένες διαχωρίζονται από την υπογραφή τους
 - Με βάση το όνομα και τους τύπους των παραμέτρων (η σειρά παίζει ρόλο)
 - Η αλλαγή απλά στον τύπο του επιστρεφόμενου αποτελέσματος δεν είναι υπερφόρτωση!!! (αλλά λάθος μεταγλώττισης)



Πρόσθετο Υλικό

- Μελετήστε και τα παραδείγματα από το **Κεφάλαιο 6** του βιβλίου:
«C++ How to Program, 9/e Paul & Harvey Deitel»
http://media.pearsoncmg.com/ph/esm/deitel/cpp_hpt_9/code_examples/Code_Examples.zip



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.0.1



Σημείωμα Αναφοράς

Copyright: Πανεπιστήμιον Πατρών, Ιωάννης Χατζηλυγερούδης, 2015.
«Οντοκεντρικός Προγραμματισμός». Έκδοση: 1.0.1 Πάτρα 2015. Διαθέσιμο
από τη δικτυακή διεύθυνση:

<https://eclass.upatras.gr/courses/CEID1105/>



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.

Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



Σημείωμα Χρήσης Έργων Τρίτων

- Οι διαφάνειες βασίζονται στο βιβλίο «C++ How to Program, 8th Edition, Harvey M. Deitel, Paul J. Deitel, Prentice Hall.»

