



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Οντοκεντρικός Προγραμματισμός

Ενότητα 5: Η ΓΛΩΣΣΑ C++

Δομές Ελέγχου

ΔΙΔΑΣΚΟΝΤΕΣ: Ιωάννης Χατζηλυγερούδης, Χρήστος
Μακρής

Πολυτεχνική Σχολή

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Δομές Ελέγχου

Εισαγωγή

- Πριν γράψουμε ένα πρόγραμμα
 - Κατανοούμε πλήρως το πρόβλημα
 - Σχεδιάζουμε προσεκτικά την παρεχόμενη λύση
- Όσο γράφουμε το πρόγραμμα
 - Γνωρίζουμε ποια δομικά μέρη είναι διαθέσιμα
 - Χρησιμοποιούμε σωστές αρχές προγραμματισμού

The art of computer programming του Donald E. Knuth



Λέξεις Κλειδιά

Κοινά με την C

Auto	break	case	char	const
Continue	default	do	double	else
Enum	extern	float	for	goto
If	int	long	register	return
Short	signed	sizeof	static	struct
Switch	typedef	union	unsigned	void
Volatile	while			

Μόνο στην C++

Asm	bool	catch	class	const_cast
delete	dynamic_cast	explicit	false	friend
inline	mutable	namespace	new	operator
private	protected	public	reinterpret_cast	true
static_cast	template	this	throw	virtual
try	typeid	typename	using	
wchar_t				



Casting

```
#include <iostream>
#include <iomanip>
using std::cout;
using std::endl;
using std::fixed;
using std::setprecision;
int main() {
    int totalAge = 75;
    int numPeople = 4;
    double averageAge = totalAge / numPeople;
    double averageAge2 = static_cast< double >( totalAge ) / numPeople;

    cout << "Avarage age is " << setprecision( 2 ) << fixed << averageAge << endl
         << "Avarage age2 is " << setprecision( 2 ) << fixed << averageAge2 << endl;
    return 0;
}
```

Η `fixed` αναγκάζει την έξοδο να εκτυπωθεί σε μορφή σταθερής υποδιαστολής (όχι σε επιστημονική μορφή). Επίσης, αναγκάζει να τυπώνονται η υποδιαστολή ακολουθούμενη από απαραίτητα μηδενικά.

Η `setprecision(2)` εκτυπώνει δύο ψηφία μετά την υποδιαστολή (στρογγυλοποίηση για να ταιριάζει στην ακρίβεια). Βιβλιοθήκη: `<iomanip>`

```
Avarage age is 18.00
Avarage age2 is 18.75
```

Η `static_cast<double>()` χειρίζεται το `totalAge` ως `double` πρόσκαιρα (casting). Απαιτείται διότι η διαίρεση δύο ακεραίων αποκόβει το υπόλοιπο.



Τελεστές Ανάθεσης

μεταβλητή τελεστής = έκφραση;
(μεταβλητή = μεταβλητή τελεστής έκφραση;)

d -= 4	(d = d - 4)
e *= 5	(e = e * 5)
f /= 3	(f = f / 3)
g %= 9	(g = g % 9)



Τελεστές αύξησης και μείωσης

- Ο τελεστής αύξησης ($++$) – μπορεί να αντικαταστήσει το: $c += 1$
- Ο τελεστής μείωσης ($--$) - μπορεί να αντικαταστήσει το: $c -= 1$
 - Προ-αύξηση
 - Όταν ο τελεστής χρησιμοποιείται πριν τη μεταβλητή ($++c$ ή $--c$)
 - Η μεταβλητή αλλάζει και στη συνέχεια υπολογίζεται η έκφραση στην οποία περιλαμβάνεται
 - Μετά-αύξηση
 - Όταν ο τελεστής χρησιμοποιείται μετά τη μεταβλητή ($c++$ ή $c--$)
 - Εκτελείται η έκφραση στην οποία περιλαμβάνεται η μεταβλητή και στη συνέχεια η μεταβλητή λαμβάνει τη νέα τιμή.



Τελεστές αύξησης και μείωσης

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

int main(){
    int x= 1;
    cout << x; // 1
    cout << x++; // 1 Πρώτα εκτύπωση μετά αύξηση
    cout << x << endl; // 2

    x = 1;
    cout << x; // 1
    cout << --x; // 0 Πρώτα αφαίρεση μετά εκτύπωση
    cout << x; // 0
    return 0;
}
```

```
112
100
```



Δομή Ελέγχου switch

```
#include <iostream>
using std::cout;
using std::cin;
int main(){
    int num;
    std::cin >> num;
    switch ( num ) {
        case 1: cout << " value 1";
            break;
        case 2:
        case 3:
            cout << " value 2 or 3";
            break;
        case (2+2):
            cout << " value 4";
        default:
            cout << " - other value.";
    }
    return 0;
}
```

Με το break συνεχίζει μετά το τέλος του switch. Το βάζουμε στο τέλος κάθε περίπτωσης ώστε να μην ελέγξει και τις επόμενες.

1
value 1

4
value 4 - other value



do/while δομή επανάληψης

- Όμοια με τη δομή **while**
 - Δημιουργεί ένα βρόγχο που ελέγχεται στο τέλος και όχι στην αρχή
 - Οι δηλώσεις στο σώμα εκτελούνται τουλάχιστον μία φορά
- Μορφή

```
do {  
    δήλωση  
} while ( συνθήκη );
```



break και continue

- Δήλωση: **break**
 - Άμεση έξοδο από **while**, **for**, **do/while**, **switch**
 - Το πρόγραμμα συνεχίζει με την πρώτη δήλωση μετά τη δομή
- Χρησιμοποιείται σε περιπτώσεις όπως:
 - Νωρίτερη διαφυγή από το βρόγχο
 - Παράβλεψη του υπόλοιπου **switch**



break και continue

- **Δήλωση: `continue`**
 - Χρησιμοποιείται στις `while`, `for`, `do/while`
 - Διαφυγή από το υπόλοιπο βρόγχο
 - Προχωρά στην επόμενη επανάληψη του βρόγχου
- **`while` και `do/while` δομές**
 - Ο έλεγχος για συνέχεια της επανάληψης ελέγχεται αμέσως μετά τη δήλωση `continue`
- **`for` δομή**
 - Εκτελείται αύξηση της έκφρασης
 - Στη συνέχεια, ελέγχεται αν θα συνεχιστεί ο βρόγχος



Παράδειγμα continue

```
for ( int x = 1; x <= 10; x++ ) {  
    if ( x % 2 )  
        continue;  
    cout << x << " ";  
}
```

2 4 6 8 10



Τελεστές Σύγκρισης (στις συνθήκες)

- $>$ μεγαλύτερο
- $<$ μικρότερο
- $>=$ μεγαλύτερο ίσο
- $<=$ μικρότερο ίσο
- $==$ ίδια τιμή
- $!=$ διαφορετική τιμή



Λογικοί Τελεστές

- Χρησιμοποιούνται ως συνθήκες σε βρόγχους και δηλώσεις `if`

- **&&** (λογικό **ΚΑΙ**)

- **true** αν και οι δύο συνθήκες είναι **true**

```
if ( gender == 1 && age >= 65 )  
    ++seniorFemales;
```

- **||** (λογικό **Ή**)

- **true** αν μία από τις δύο συνθήκες είναι **true**

```
if ( semesterAverage >= 90 || finalExam >= 90 )  
    cout << "Student grade is A" << endl;
```



Λογικοί Τελεστές

- Χρησιμοποιούνται ως συνθήκες σε βρόγχους και δηλώσεις `if`

- **&&** (λογικό **ΚΑΙ**)

- **true** αν και οι δύο συνθήκες είναι **true**

```
if ( gender == 1 && age >= 65 )  
    ++seniorFemales;
```

- **||** (λογικό **Ή**)

- **true** αν μία από τις δύο συνθήκες είναι **true**

```
if ( semesterAverage >= 90 || finalExam >= 90 )  
    cout << "Student grade is A" << endl;
```



Λογικοί Τελεστές

- ! (λογικό **NOT**)

- Επιστρέφει **true** όταν η συνθήκη είναι **false**, & αντιστρόφως

```
if ( !( grade == sentinelValue ) )  
    cout << "The next grade is " <<  
grade << endl;
```

Εναλλακτικά:

```
if ( grade != sentinelValue )  
    cout << "The next grade is " <<  
grade << endl;
```



Ισότητα (==) και Ανάθεση (=)

- Κοινό λάθος
 - Δεν επιστρέφεται συνήθως συντακτικό λάθος
- Όψεις του προβλήματος
 - Οι εκφράσεις που έχουν τιμή μπορούν να χρησιμοποιηθούν για να ληφθεί απόφαση
 - Zero = false, nonzero = true
 - Οι δηλώσεις ανάθεσης παράγουν μία τιμή (αυτή που αναθέτουν)



Πρόσθετο Υλικό

- Μελετήστε και τα παραδείγματα από τα **Κεφάλαια 4, 5** του βιβλίου:
«C++ How to Program, 9/e Paul & Harvey Deitel»
[http://media.pearsoncmg.com/ph/esm/deitel/cpp htp_9/code_examples/Code_Examples.zip](http://media.pearsoncmg.com/ph/esm/deitel/cpp_htp_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.»

