«Προγραμματισμός Ι»

1- Τα Βασικά της C

Δρ. Χαράλαμπος Καρανίκας
Π.Σ. Μηχανικών Πληροφορικής

ΤΑ ΣΥΣΤΑΤΙΚΑ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΤΗΣ C

- Δομικό στοιχείο ενός προγράμματος C είναι οι συναρτήσεις
- Κάθε μία από αυτές περιέχει 1 ή περισσότερες εντολές
- Οι εντολές τερματίζονται με «;»
- Η γενική μορφή μίας συνάρτησης της C:

```
τύπος-επιστρεφόμενης τιμής όνομα-συνάρτησης(λίστα-παραμέτρων)
{
κορμός-συνάρτησης (εντολές)
}
```

- Αποδεκτοί τύποι δεδομένων
- Αποδεκτή ονοματολογία συναρτήσεων
- Κάθε πρόγραμμα C περιέχει μια συνάρτηση main() από την οποία ξεκινά η εκτέλεση του προγράμματος
- Ο κορμός ενός προγράμματος και οι συναρτήσεις Βιβλιοθήκης

ΤΑ ΣΥΣΤΑΤΙΚΑ ΕΝΟΣ ΠΡΟΓΡΑΜΜΑΤΟΣ ΤΗΣ C (συνέχεια)

- Συναρτήσεις εισόδου/εξόδου
  π.χ. printf(“αλφαριθμητικό”);
- Οι ντιρεκτίβες του προεπεξεργαστή της C
- Το πρώτο πρόγραμμα σε C:

```
#include <stdio.h>

int main(void)
{
        printf("This is a short C program.");

        return 0;
}
```

```
#include <stdio.h>

int main(void)
{
        printf("This is ");
        printf("another C ");
        printf("program.");

      return 0;
}
```

Με την οδηγία #include <όνομα_αρχείου> ο μεταγλωττιστής υποχρεώνεται να ενσωματώσει και να συμπεριλάβει (include) τα περιεχόμενα του αρχείου στον κώδικα του προγράμματος.
Το αρχείο stdio.h (standard input output) περιέχει τις βασικές (standard) δηλώσεις των συναρτήσεων με τις οποίες γίνεται εμφάνιση δεδομένων στην οθόνη (output) και εισαγωγή δεδομένων από το πληκτρολόγιο (input).

ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ ΚΑΙ ΕΚΧΩΡΗΣΗ ΤΙΜΩΝ

- Μεταβλητή είναι μία επώνυμη θέση στην μνήμη
- Στην C αποτελεί νόμο: «Δήλωση πριν την Χρήση»
- Η C υποστηρίζει πέντε βασικούς τύπους δεδομένων:

Τύπος	Δεσμευμένη λέξη
χαρακτήρας	char
προσημασμένος ακέραιος	int
αριθμός κινητής υπο/λής	float
αρ. κιν. υπο/λής, διπλής ακρίβειας	double
απουσία τιμής	void

Δήλωση (αποτελεί εντολή): τύπος όνομα-μεταβλητής;

Παράδειγμα: int counter;

- Θέσεις δήλωσης μεταβλητών & προσπέλαση τους
  1. Μέσα σε συναρτήσεις (local variables)
  2. Έξω από όλες τις συναρτήσεις (global variables)
- Δημιουργία κατά την κλήση – Καταστροφή κατά τον τερματισμό

ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ ΚΑΙ ΕΚΧΩΡΗΣΗ ΤΙΜΩΝ (συνέχεια)

- Πολλαπλή δήλωση μεταβλητών

  Π.χ. float x, y, z;

- Ονοματολογία μεταβλητών όμοια με συναρτήσεις (0-9 όχι στην αρχή, γράμματα αλφαβήτου και κάτω παύλα)
- Η C είναι case-sensitive

  Π.χ. int count, COUNT;

- Εκχώρηση τιμών (αποτελεί εντολή): όνομα-μεταβλητής = τιμή;

Παράδειγματα:
```
counter = 100;
fl = 100.1;
fl = 100.0;
```

- Κάτι ακόμα για τη συνάρτηση εξόδου printf()

```
printf("This prints the number %d", 99);
printf("This displays %d, too", 99);
```

- % - Κωδικός μορφοποίησης

  Άλλοι κωδικοί: %c για χαρακτήρες & %f για αριθμούς κινητής υποδιαστολής

ΔΗΛΩΣΗ ΜΕΤΑΒΛΗΤΩΝ ΚΑΙ ΕΚΧΩΡΗΣΗ ΤΙΜΩΝ
(συνέχεια)

```
#include <stdio.h>

int main(void)
{
  int num;

  num = 100;
  printf("The value is %d", num);

  return 0;
}
```

```
#include <stdio.h>

int main(void)
{
  char ch;
  float f;
  double d;

  ch = 'X';
  f = 100.123;
  d = 123.009;

  printf("ch is %c, ", ch);
  printf("f is %f, ", f);
  printf("d is %f", d);

  return 0;
}
```

ΕΙΣΟΔΟΣ ΑΡΙΘΜΩΝ ΑΠΟ ΤΟ ΠΛΗΚΤΡΟΛΟΓΙΟ

- scanf(“%d”, &όνομα-μεταβλητής-int);

  Π.χ. int num;
  scanf("%d", &num);

- scanf(“%f”, &όνομα-μεταβλητής-float);

  Π.χ. float fl;
  scanf("%f", &fl);

- scanf(“%lf”, &όνομα-μεταβλητής-double);

  Π.χ. double dbl;
  scanf("%lf", &dbl);

```
#include <stdio.h>

int main(void)
{
  int num;
  float f;

  printf("Enter an integer: ");
  scanf("%d", &num);

  printf("Enter a floating point number: ");
  scanf("%f", &f);

  printf("%d ", num);
  printf("%f", f);

  return 0;
}
```

ΕΚΤΕΛΕΣΗ ΥΠΟΛΟΓΙΣΜΩΝ ΜΕ ΑΡΙΘΜΗΤΙΚΕΣ ΕΚΦΡΑΣΕΙΣ

- Η C ορίζει ένα εκτεταμένο σύνολο αριθμητικών τελεστών:

Τελεστής	Σημασία
+	πρόσθεση
-	αφαίρεση & μοναδιαίος τελεστής
*	πολλαπλασιασμός
/	διαίρεση
%	υπόλοιπο διαίρεσης ακεραίων

- Εκφράσεις στη δεξιά πλευρά εντολών εκχώρησης

  Π.χ. int answer;
  answer = 100 * 31;

- Προτεραιότητα πράξεων: Οι τελεστές *, /, % έχουν υψηλότερη προτεραιότητα από τους +, -
- Χρησιμοποιείτε παντού παρενθέσεις!

  Π.χ. 10 – 2 * 5
  (10 – 2) * 5

- Μία έκφραση μπορεί να περιέχει μεταβλητές, σταθερές ή συνδυασμό

  Π.χ. answer = count - 100;

ΕΚΤΕΛΕΣΗ ΥΠΟΛΟΓΙΣΜΩΝ ΜΕ ΑΡΙΘΜΗΤΙΚΕΣ ΕΚΦΡΑΣΕΙΣ (συνέχεια)

Παράδειγμα 1:

```
#include <stdio.h>

int main(void)
{
  printf("%d", 5/2);
  printf(" %d", 5%2);
  printf(" %d", 4/2);
  printf(" %d", 4%2);

  return 0;
}
```

Παράδειγμα 2:

```
count *num+88/val-19%count

(count * num) + (88 / val) - (19 %
count)
```

ΕΚΤΕΛΕΣΗ ΥΠΟΛΟΓΙΣΜΩΝ ΜΕ ΑΡΙΘΜΗΤΙΚΕΣ ΕΚΦΡΑΣΕΙΣ (συνέχεια)

Παράδειγμα 3:

```
#include <stdio.h>

int main(void)
{
  int len, width;

  printf("Enter length: ");
  scanf("%d", &len);
  printf("Enter width: ");
  scanf("%d", &width);

  printf("Area is %d", len * width);

  return 0;
}
```

Παράδειγμα 4:

```
#include <stdio.h>

int main(void)
{
  int i;

  i = 10;
  i = -i;
  printf("This is i: %d", i);

  return 0;
}
```

ΠΡΟΣΘΗΚΗ ΣΧΟΛΙΩΝ ΣΕ ΕΝΑ ΠΡΟΓΡΑΜΜΑ

```
/* This is a comment. */

/*
  This is a longer comment
  that extends over
  five lines.
*/

/* this is a comment /* this is another
      comment nested inside the first --
      which will cause a syntax error */
      with a nested comment
*/

pri/* wrong */ntf("this won't work");
```

```
/* This program converts earth days into Jovian years.
https://www.exploratorium.edu/ronh/age/ */

#include <stdio.h>

int main(void)
{
  float e_days; /* number of earth days */
  float j_years; /* equivalent number of Jovian years */

  /* get number of earth days */
  printf("Enter number of earth days: ");
  scanf("%f", &e_days);

  /* now, compute Jovian years */
  j_years = e_days / (365.0 * 12.0);

  /* display the answer */
  printf("Equivalent Jovian years: %f", j_years);

  return 0;
}
```

ΓΡΑΨΤΕ ΤΙΣ ΔΙΚΕΣ ΣΑΣ ΣΥΝΑΡΤΗΣΕΙΣ

```
#include <stdio.h>

void func1(void);  /* prototype for func1()*/

int main(void)
{
        printf("I ");
        func1();
        printf("C.");

        return 0;
}

void func1(void)
{
        printf("like ");
}
```

```
/* This program has three functions. */
#include <stdio.h>

void func1(void); /* prototypes */
void func2(void);

int main(void)
{
        func2( );
        printf("3");
        return 0;
}

void func2(void) {
        func1();
        printf("2 ");
}
void func1(void) {
         printf("1 ");
}
```

ΧΡΗΣΗ ΣΥΝΑΡΤΗΣΕΩΝ ΓΙΑ ΤΗΝ ΕΠΙΣΤΡΟΦΗ ΤΙΜΩΝ - 1

```c
#include <stdio.h>
#include <math.h> /* needed by sqrt() */

int main(void)
{
    double answer;

    answer = sqrt(10.0);
    printf("%f", answer);

    return 0;
}
```

```c
#include <stdio.h>
#include <math.h>
/* needed by sqrt() */

int main(void)
{
    printf("%f", sqrt(10.0));

    return 0;
}
```

```c
#include <stdio.h>

int func(void); /* prototype */

int main(void)
{
    int num;

    num = func( );
    printf("%d", num);

    return 0;
}

int func(void)
{
    return 10;
}
```

Εντολή Return

```
#include <stdio.h>

void func1(void);

int main(void)
{
        func1();

        return 0;
}

void func1(void)
{
        printf("This is printed.");
        return; /* return with no value */
        printf("This is never printed.");
}
```

- Στην εντολή **return**, η συνάρτηση επιστρέφει (τερματίζει) αμέσως.
- Καμία εντολή μετά δεν εκτελείται.
- Η τιμή της return μπορεί να είναι οποιαδήποτε έγκυρη έκφραση της C.
- Χωρίς τιμή η **Return ;** Κυρίως σε συναρτήσεις τύπου **void.**

ΧΡΗΣΗ ΣΥΝΑΡΤΗΣΕΩΝ ΓΙΑ ΤΗΝ ΕΠΙΣΤΡΟΦΗ ΤΙΜΩΝ - 2

```
#include <stdio.h>

int get_sqr(void);

int main(void)
{
        int sqr;

        sqr = get_sqr( );
        printf("Square: %d", sqr);

        return 0;
}

int get_sqr(void)
{
        int num;

        printf("Enter a number: ");
        scanf("%d", &num);
        return num*num; /* square the number */
}
```

ΧΡΗΣΗ ΟΡΙΣΜΑΤΩΝ ΣΕ ΣΥΝΑΡΤΗΣΕΙΣ - 1

- *Παραμετρικές συναρτήσεις.* Οι συναρτήσεις που δέχονται ορίσματα.
- *Όρισμα* μιας συνάρτησης είναι μια τιμή η οποία περνιέται στην συνάρτηση όταν την καλείται.
- Μηδέν, ένα ή περισσότερα ορίσματα.(Πρότυπο ANSI C τουλάχιστον 31 ορίσματα)
- *Τυπικές παράμετροι.* Δήλωση ειδικών μεταβλητών για να λαμβάνουν τις τιμές των ορισμάτων.

```
void sum(int x, int y)
{
      printf("%d ", x + y);
}
```

ΧΡΗΣΗ ΟΡΙΣΜΑΤΩΝ ΣΕ ΣΥΝΑΡΤΗΣΕΙΣ - 2

```
/* Χρήση της συνάρτησης sum*/
#include <stdio.h>

void sum(int x, int y);

int main(void)
{
        sum(1, 20);
        sum(9, 6);
        sum(81, 9);

        return 0;
}

void sum(int x, int y)
{
        printf("%d ", x + y);
}
```

```
/* Συνάρτηση outchar() για την έξοδο
χαρακτήρων στην οθόνη */
#include <stdio.h>

void outchar(char ch);

int main(void)
{
        outchar('A');
        outchar('B');
        outchar('C');

        return 0;
}

void outchar(char ch)
{
        printf("%c", ch);
}
```