

Εργαστήριο 3ο

Χρήση έτοιμων συναρτήσεων - Δημιουργία συναρτήσεων

Κλήση με τιμή / αναφορά - Προκαθορισμένες τιμές

Υπερφόρτωση συναρτήσεων

Άσκηση 1 (βασικά των συναρτήσεων)

Η ελκτική δύναμη που δημιουργείται μεταξύ δύο σωμάτων δίνεται από την σχέση

$$F = G \frac{m_1 m_2}{d^2}$$

Όπου m_1 και m_2 είναι οι μάζες των δύο σωμάτων, d η μεταξύ τους απόσταση και $G = 6,67 \times 10^{-11} N \frac{m^2}{kg^2}$ είναι η σταθερά της βαρύτητας.

Να δημιουργήσετε μια συνάρτηση που να υπολογίζει και να επιστρέφει την δύναμη που ασκείται μεταξύ δύο σωμάτων. Στην συνάρτηση αυτή να υπάρχει προκαθορισμένη παράμετρος για την m_2 η μάζα της γής $M = 5,972 \times 10^{24} kg$. Πιο συγκεκριμένα να δημιουργήσετε τον παρακάτω κώδικα με όνομα Lab3E1

```
#include "stdafx.h"
#include <iostream>
using namespace std;

double calc_attr_force(double m1, double m2, double d);

int main()
{
    double mash1, mash2, dist, force;
    cout << "Give the masses of the two elements(in kilograms): ";
    cin >> mash1 >> mash2;
    cout << "Give the distance of the two elements (in meters): ";
    cin >> dist;
    force = calc_attr_force(mash1, mash2, dist);
    cout << "The force between the two elements is " << force << " Newtons" << endl;
    return 0;
}

double calc_attr_force(double m1, double m2, double d) {
    const double g = 6.67e-11;
    double tmp;
    tmp = g * m1 * m2 / d * d;
    return tmp;
}
```

Να απαντήσετε στις παρακάτω ερωτήσεις:

1. Ποια είναι η δήλωση (declaration) και ποιος ο ορισμός (definition) της συνάρτησης και ποια η κλήση (call) της συνάρτησης; Γιατί χρειάζεται το declaration;
2. Ποια είναι το α) όνομα (name) β) τύπος επιστροφής (return type) γ) παράμετροι (parameters) της συνάρτησης δ) ορίσματα (arguments) της συνάρτησης;
3. Πόσες μπορεί να είναι οι παράμετροι (parameters) μιας συνάρτησης; Πόσες οι τιμές επιστροφής (return values);

4. Έχει σημασία η σειρά με την οποία δίνουμε τα ορίσματα (arguments) κατά την κλήση μιας συνάρτησης;

Δοκιμάστε τις παρακάτω αλλαγές

1. Δοκιμάστε να αλλάξετε το mash2 έτσι ώστε να μην είναι μια μεταβλητή που παίρνει τιμή από το πληκτρολόγιο αλλά να είναι μια σταθερά (π.χ. η μάζα της γης 5,972e24 kg). Υπάρχει πρόβλημα που το όρισμα στην κλήση μιας συνάρτησης είναι σταθερή τιμή;
2. Δοκιμάστε στον ορισμό της συνάρτησης να διαγράψετε την προσωρινή μεταβλητή tmp και να επιστρέψετε με την return ολόκληρη την έκφραση. Υπάρχει πρόβλημα;
3. Δοκιμάστε να αλλάξετε τα ονόματα των μεταβλητών mash1, mash2 σε m1,m2. Υπάρχει πρόβλημα που έχουν το ίδιο όνομα με τις παραμέτρους (parameters) της συνάρτησης; Αν αλλάζαμε την τιμή των m1 ή m2 μέσα στην συνάρτηση αυτό θα επηρέαζε την τιμή των μεταβλητών στην main;
4. Δημιουργήστε μια double μεταβλητή tmp μέσα στην main και αρχικοποιείστε την με την τιμή 10,45. Στο τέλος του προγράμματος ποια τιμή θα έχει η tmp;

Άσκηση 2 (κλήση με αναφορά)

Να δημιουργήσετε ένα project Lab3E2 το οποίο περιέχει μια συνάρτηση που να δέχεται τις δύο ακέραιες μεταβλητές και να αντιστρέφει τις τιμές τους.

```
#include <iostream>
using namespace std;
void swap(int &x, int &y);

int main() {
    int a, b;
    cout << "Give me two integers: ";
    cin >> a >> b;
    swap(a, b);
    cout << "The first var has value " << a << endl;
    cout << "The second var has value " << b << endl;
}

void swap(int &x, int &y) {
    int tmp;
    tmp = x;
    x = y;
    y = tmp;

    return;
}
```

Απαντήστε στις παρακάτω ερωτήσεις:

1. Σε ποιο/α από τα τρία έχουν γίνει αλλαγές α) declaration b) definition c) function call;
2. Ποια είναι η αλλαγή που έχει γίνει και ποια είναι η διαφορά στην λειτουργία του προγράμματος που αυτή προκαλεί

Άσκηση 3 (Υπερφόρτωση overloading συναρτήσεων)

Να δημιουργήσετε μια συνάρτηση weight, για μια εταιρία που παρασκευάζει κουτιά σε σχήμα κύβου, και υπολογίζει το βάρος του κάθε κουτιού. Σαν παράμετρο θα δέχεται την ακμή του κύβου σε εκατοστά και το βάρος του χαρτιού ανά τετραγωνικό εκατοστό.

Η εταιρία αυτή αποφασίζει να κατασκευάσει και κουτιά σε σχήμα παραλληλόγραμμου. Να δημιουργήσετε μια συνάρτηση `weight` που να δέχεται σαν είσοδο τις τρεις διαστάσεις του κουτιού (πλάτος, μήκος, ύψος) και το βάρος του χαρτιού ανά τετραγωνικό μέτρο. Πιο συγκεκριμένα να δημιουργήσετε ένα project με το όνομα Lab3E3.

```
#include<iostream>
using namespace std;

double calc_weight(int acme, double weight);
double calc_weight(int l, int w, int h, double weight);

int main() {
    const double weight_per_cm = 3.44;
    int x1;
    double weight_box;

    cout << "Give me the acme of the box: ";
    cin >> x1;
    weight_box = calc_weight(x1, weight_per_cm);
    cout << "The cube box weights " << weight_box<<endl<<endl;

    int y1, y2,y3;
    double weight_new_box;
    cout << "Give me the three dimensions of the box:";
    cin >> y1 >> y2 >> y3;
    weight_new_box = calc_weight(y1, y2, y3, weight_per_cm);
    cout << "The new box weights " << weight_new_box << endl;

    return 0;
}

double calc_weight(int acme, double weight) {
    int surface;
    surface = 6 * acme * acme;

    return surface * weight;
}

double calc_weight(int l, int w, int h, double weight) {
    int surface;
    surface = 2 * (l*w + w*h + h*l);

    return surface * weight;
}
```

Άσκηση 4

Να δημιουργήσετε μια συνάρτηση που θα δέχεται τέσσερις ακέραιους αριθμούς και θα επιστρέφει την τυπική απόκλιση. Η τυπική απόκλιση υπολογίζεται σύμφωνα με τον τύπο (όπου μ είναι η μέση τιμή των αριθμών αυτών)

$$s = \sqrt{\frac{1}{N} \sum_{i=1}^N (x^i - \mu)^2}$$

Ακόμα θα γράψετε ένα πρόγραμμα που θα ζητάει από τον χρήστη τέσσερις αριθμούς από το πληκτρολόγιο και θα υπολογίζει και θα τυπώνει στην οθόνη την τυπική απόκλιση κάνοντας κλήση της συνάρτησης που δημιουργήσατε.

Τροποποιήστε την παραπάνω συνάρτηση ώστε να επιστρέφει και τον μέσο όρο σαν παράμετρο

Άσκηση 5

Να δημιουργήσετε μια συνάρτηση `sort3` που να δέχεται σαν παραμέτρους τρεις ακεραίους αριθμούς. Στην συνέχεια η συνάρτηση αυτή θα αντιμετωπίζει τις τιμές των παραμέτρων έτσι ώστε να είναι ταξινομημένες με αύξουσα σειρά (δηλαδή η μικρότερη τιμή θα ανατίθεται στην πρώτη παράμετρο, η αμέσως μεγαλύτερη στην δεύτερη παράμετρο και η μεγαλύτερη από όλες στην τρίτη παράμετρο).

Να δημιουργήσετε ένα πρόγραμμα που θα ζητάει από τον χρήστη τρεις ακεραίους αριθμούς θα τους αναθέτει σε τρεις μεταβλητές και μετά θα τους τυπώνει. Στην συνέχεια θα καλεί την συνάρτηση δίνοντας σαν παραμέτρους τις τρεις αυτές μεταβλητές και θα ξανατυπώνει τις τιμές των μεταβλητών.

Άσκηση 6

Να γράψετε ένα πρόγραμμα που θα υπολογίζει την επιφάνεια και τον όγκο είτε για σφαίρα είτε για κύλινδρο είτε για κώνο. Το πρόγραμμα θα λειτουργεί ως εξής.

Θα ρωτάει τον χρήστη αν θέλει να υπολογίζει για σφαίρα, κύλινδρο ή κώνο. Στην πρώτη περίπτωση θα ρωτάει μόνο την ακτίνα της σφαίρας ενώ στις άλλες δύο περιπτώσεις θα ρωτάει την ακτίνα της βάσης και το ύψος. Στην συνέχεια θα εμφανίζει την επιφάνεια και το όγκο του σχήματος. Τέλος θα ρωτάει τον χρήστη αν θέλει να εκτελέσει και καινούργιο υπολογισμό.

ΠΡΟΣΟΧΗ! Ο υπολογισμός της επιφάνειας και του όγκου θα γίνεται με την κατάλληλη συνάρτηση. Θα δημιουργηθούν 3 συναρτήσεις με το όνομα `calc_surface` και 3 με το όνομα `calc_value`

$$\text{Επιφάνεια Σφαίρας } S = 4\pi r^2 \quad \text{Όγκος Σφαίρας } V = \frac{4}{3}\pi r^3$$

$$\text{Επιφάνεια Κυλίνδρου } S = 2\pi r(h + r) \quad \text{Όγκος Κυλίνδρου } V = \pi r^2 h$$

$$\text{Επιφάνεια Κώνου } S = \pi r(\sqrt{r^2 + h^2} + r) \quad \text{Όγκος Κώνου } V = \frac{1}{3}\pi r^2 h$$

Άσκηση 6

Να δημιουργήσετε μια συνάρτηση `find_digit` που να δέχεται σαν είσοδο έναν ακέραιο αριθμό `num` και έναν μικρό ακέραιο αριθμό `n`. Η συνάρτηση θα επιστρέφει το ψηφίο που βρίσκεται στην θέση `n` του αριθμού `num`. Αν ο αριθμός `num` δεν έχει `n` ψηφία τότε θα επιστρέφει την τιμή `-1`.

Να δημιουργήσετε μία συνάρτηση `main` που να ελέγχει την συνάρτηση που δημιουργήσατε.