Δομή προγράμματος, Είσοδος – Έξοδος Δεδομένων

Περιεχόμενα

Η δομή του προγράμματος στη γλώσσα προγραμματισμού c++ 1
Εντολές Εξόδου 2
Μορφοποίηση Εξόδου 3
Είσοδος δεδομένων 5
Σταθερές 6
Αριθμητικοί τελεστές (+ , - , * , / , %) 6
Προσαρμογή τύπου - Casting 8
Βιβλιοθήκη cmath 9
Λύσεις ασκήσεων 9

Η δομή του προγράμματος στη γλώσσα προγραμματισμού c++.

Ένα πρόγραμμα στη γλώσσα προγραμματισμού C++ έχει την εξής δομή:

```
#include <iostream>
using namespace std;
int main(){
  // Εδώ γράφουμε τις εντολές του προγράμματος μας
   return 0;
}
```

Ας αναλύσουμε μαζί αυτό το πρόγραμμα :

- **#include <iostream>**
  Οδηγία ώστε το πρόγραμμα να συμπεριλάβει τη βιβλιοθήκη **<iostream>** για να μπορεί να εκτελεί εντολές εισόδου και εξόδου. Σε κατοπινό στάδιο θα δούμε εισαγωγή και άλλων βιβλιοθηκών, όπως η **cmath**.
- **using namespace std;**
  Όλες οι βασικές βιβλιοθήκες της **C++** είναι δηλωμένες μέσα στο namespace **std**. Χωρίς αυτή την εντολή θα πρέπει κάθε φορά να δηλώνουμε το **namespace**.
- **int main(){**
  // Εδώ γράφουμε τις εντολές του προγράμματος μας
  return 0;
  }
  Η κύρια συνάρτηση του προγράμματος είναι η **main** και μέσα σε αυτή γράφουμε τις εντολές του προγράμματός μας. Η εντολή **return 0;** απλά πληροφορεί το λειτουργικό σύστημα ότι το πρόγραμμα έτρεξε και τερμάτισε χωρίς κάποιο πρόβλημα.

  Προσοχή: Δεν ξεχνούμε ποτέ τις παρενθέσεις της **main**, ούτε τα άγκιστρα, μέσα στα οποία γράφουμε τις εντολές του προγράμματος μας.

Σημαντικό: Οι δύο πλάγιες γραμμές **//** δηλώνουν σχόλια. Τα σχόλια απλά βοηθούν τον προγραμματιστή να θυμάται ή να κατανοεί ένα πρόγραμμα και δεν λαμβάνονται υπόψη από τη γλώσσα προγραμματισμού.

Σημαντικό: Κάθε εντολή τελειώνει πάντοτε με το ελληνικό ερωτηματικό **; Ποτέ δεν το ξεχνούμε αυτό**

Εντολές Εξόδου

Τώρα που έχουμε δει τη βασική δομή του προγράμματος ας δούμε πως γράφουμε την εντολή εξόδου, δηλαδή πως παρουσιάζουμε κάτι στην οθόνη.

Αυτό μπορεί να γίνει με τη χρήση της συνάρτησης **cout** και του τελεστή **<<**.

Παράδειγμα

```
#include <iostream>
using namespace std;
int main()
{
  cout<<"I love c++";
  return 0;
}
```

Ο τελεστής εξαγωγής **<<** στέλνει οτιδήποτε βρίσκεται στα δεξιά του στην οθόνη.

Σημαντικό: Το κείμενο πάντοτε περικλείεται σε διπλά εισαγωγικά:

Παράδειγμα

```
#include <iostream>
using namespace std;
int main()
{
  cout<<"I love c++";
  return 0;
}
```

Παράδειγμα

```
#include <iostream>
using namespace std;
int main()
{
  cout<<"I love"<<" c++";
  return 0;
}
```

Η αλλαγή γραμμής μπορεί να γίνει με τη χρήση της εντολής **endl**.

Παράδειγμα

```
#include <iostream>
using namespace std;
int main()
{
  cout<<"I love"<<endl<<"c++";
  return 0;
}
```

Ώρα για άσκηση:

Άσκηση 1.1
Να γράψετε το πρόγραμμα που να τυπώνει στην οθόνη το επόμενο τρίγωνο με αστεράκια.
```
*
**
***
****
*****
******
```

Μορφοποίηση Εξόδου

Για να μορφοποιήσουμε τα δεδομένα εξόδου πρέπει να εισάγουμε στο πρόγραμμα μας τη βιβλιοθήκη **iomanip.**
Το **setw** καθορίζει τα διαστήματα εκτύπωσης στην οθόνη.
Το **setprecision** καθορίζει τα δεκαδικά ψηφία σε πραγματικούς αριθμούς. Είναι απαραίτητο να γράψουμε την εντολή **fixed** πριν από την εντολή **setprecision**, αν θέλουμε να έχουμε ακριβώς τον αριθμό των ψηφίων που ζητούμε.

Παράδειγμα

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   cout<<"Keep"<<setw(8)<<"calm"<<endl;
   cout<<fixed<<setprecision(3)<<1.4463<<endl;
  return 0;
}
```

Αυτό το πρόγραμμα θα παρουσιάσει τα επόμενα. Τα κενά διαστήματα συμβολίζονται με □
Keep□□□□calm
1.446
Σημαντικό: Όταν χρησιμοποιούμε το **setw** για να παρουσιάσουμε πραγματικό αριθμό, τότε ως διάστημα εκτύπωσης μετρά και η τελεία.

Παράδειγμα

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   cout<<setw(6)<<fixed<<setprecision(3)<<2.86<<endl;
  return 0;
}
```

Σε αυτό το παράδειγμα ζητούμε να παρουσιαστεί ο αριθμός **2.86 με 3 δεκαδικά** και να καταλάβει **6 διαστήματα**. Το αποτέλεσμα είναι **□2.860.**

Σημαντικό: Το **setprecision** εκτελεί στρογγυλοποίηση αν το επόμενο δεκαδικό ψηφίο είναι μεγαλύτερο ή ίσο με το 5.

Παράδειγμα

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   cout<<setw(6)<<"Keep"<<setw(8)<<"calm"<<endl;
   cout<<setw(8)<<fixed<<setprecision(2)<<1.4463<<endl;
  return 0;
}
```

□□**Keep**□□□□**calm**

1.45

Προσοχή: Αν το **setw** καθορίσει διαστήματα εκτύπωσης λιγότερα από όσο απαιτεί η εκτύπωση, τότε απλά γίνεται η εκτύπωση χωρίς να λαμβάνεται υπόψη το **setw**.

Παράδειγμα

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   cout<<setw(2)<<"Dog"<<setw(3)<<1234<<endl;
  return 0;
}
```

Το αποτέλεσμα στην οθόνη είναι **Dog1234**. Το Dog χρειάζεται 3 διαστήματα για να εκτυπωθεί και το **setw** ζητά 2. Παρομοίως, για τον αριθμό **1234** που χρειάζεται **4** διαστήματα καθορίζονται μόνο **3**. Και στις δύο περιπτώσεις αγνοείται η εντολή **setw**.

Ώρα για άσκηση:

Άσκηση 1.2
Τι θα παρουσιάσει στην οθόνη το επόμενο πρόγραμμα; Τα κενά διαστήματα να τα συμβολίσετε με □

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
   cout<<setw(6)<<"This"<<setw(3)<<"is"<<setw(1)<<" a"<<" cat"<<endl;
   cout<<setw(3)<<11<<setw(2)<<44<<setw(1)<<22<<2<<endl;
   cout<<fixed<<setprecision(3)<<3.141592<<endl;
  return 0;
}
```

Είσοδος δεδομένων

Οι πληροφορίες που αλλάζουν κατά την διάρκεια της εκτέλεσης ενός προγράμματος αποθηκεύονται προσωρινά σε κάποια θέση μνήμης του υπολογιστή που ονομάζεται μεταβλητή. **Για να χρησιμοποιήσουμε μια μεταβλητή πρέπει να δηλώσουμε τον τύπο της και το όνομα της.**

Προσοχή: Το όνομα της μεταβλητής πρέπει να ξεκινά με γράμμα του λατινικού αλφαβήτου και να μην περιέχει απαγορευμένα σύμβολα όπως για παράδειγμα ^,&,= κλπ.

Ανάλογα με το είδος των δεδομένων που θα κρατά μια μεταβλητή, επιλέγουμε και τον τύπο της. Οι τύποι δεδομένων μια μεταβλητής είναι ως εξής:

- **char Χαρακτήρας -128 έως 127.**
- **short ακέραιος αριθμός -32,768 έως 32,767.**
- **int ακέραιος αριθμός -2,147,483,648 έως 2,147,483,647.**
- **long long ακέραιος αριθμός $(-2^{63}+1)$ έως $(+2^{63}+1)$.**
- **float πραγματικός αριθμός 1.2E-38 έως 3.4E+38.**
- **double πραγματικός αριθμός 2.2E-308 έως 1.8E+308.**
- **long double πραγματικός αριθμός 3.4E-4932 έως 1.2E+4932.**
- **bool λογικές τιμές.**

Μπορούμε να τοποθετήσουμε αρχικές τιμές στις μεταβλητές μας κατά τη δήλωση τους.

Παράδειγμα

```
#include <iostream>
using namespace std;
int main()
{  int x;
   int a,b;
   int w=13;
   float z=-0.4;
   char c='a';
   bool f=true;

  return 0;
}
```

Για να εισάγουμε τιμές σε ένα πρόγραμμα χρησιμοποιούμε την εντολή **cin** και τον τελεστή **>>**
Το επόμενο από παράδειγμα θα σας βοηθήσει να καταλάβετε.

Παράδειγμα

```
#include <iostream>
using namespace std;
int main()
{  int sum;
   int a,b;
   cin>>a;
   cin>>b;
   sum=a+b;
   cout<<sum;
  return 0;
}
```

Ώρα για άσκηση:

Άσκηση 1.3
Να γράψετε το πρόγραμμα που δέχεται 3 ακέραιους αριθμούς και παρουσιάζει το άθροισμα και το γινόμενο τους.
Πριν γράψετε αυτό το πρόγραμμα απαντήστε τις επόμενες ερωτήσεις.

- Πόσες τιμές πρέπει να δώσει ο χρήστης;
- Πόσες μεταβλητές πρέπει να χρησιμοποιήσετε;
- Τι θα τυπώσει το πρόγραμμα στην οθόνη;

Σταθερές

Οι σταθερές έχουν μια προκαθορισμένη τιμή κατά την διάρκεια εκτέλεσης του προγράμματος.
Υπάρχουν δύο τρόποι για να δηλώσετε σταθερές. Ο πρώτος τρόπος είναι με τη χρήση του **#define** και ο δεύτερο με τη χρήση της δεσμευμένης λέξης **const**. Το επόμενο παράδειγμα θα σας διαφωτίσει.

Παράδειγμα

```
#include <iostream>
#include <iomanip>
#include <cmath>

#define k 10     //σταθερά με όνομα k και τιμή 10
const float pi=3.141592;  //σταθερά με όνομα pi και τιμή 3.141592

using namespace std;
int main()
{  float  aktina, emvadon;
   cin>>aktina;
   emvadon=pi*aktina*aktina;
   cout<<fixed<<setprecision(3)<<emvadon;
  return 0;
}
```

Σημαντικό: Το **#define** δεν χρειάζεται **=** και ούτε **;**

Σημαντικό: Το **const** χρειάζεται **= και ;** και πριν από τη σταθερά τον **τύπο της.**

Αριθμητικοί τελεστές (+ , - , * , / , %)

Υπάρχουν οι επόμενοι μαθηματικοί τελεστές στη c++

+	πρόσθεση
-	αφαίρεση
*	πολλαπλασιασμός
/	διαίρεση
%	υπόλοιπο ακέραιας διαίρεσης

Προσοχή: Το % εφαρμόζεται μόνο σε ακέραιους αριθμούς ή μεταβλητές ακέραιου τύπου.

Παράδειγμα

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
  int a,b,c,d;
  a=12;
  b=5;
  c=6;
  d=a % 10;
  cout<<d<<endl;
  d=a / 10;
  cout<<d<<endl;
  d=b % 2;
  cout<<d<<endl;
  d=c % 2;
  cout<<d<<endl;
  cout<<1254/100<<endl;
  cout<<1234/10<<endl;
  cout<<1234/1000<<endl;
  return 0;
}
```

Η έξοδος του προγράμματος στην οθόνη είναι η επόμενη:

```
2
1
1
0
12
123
1
```

Σημαντικό: Αν ο αριθμός **b** είναι ζυγός, τότε το **b %2** δίνει απάντηση **0**. Αν περιττός, τότε η απάντηση είναι **1**.

Σημαντικό: Το **b%10** μας δίνει το τελευταίο ψηφίο του αριθμού **b**. Το **b/10** μας επιστρέφει τον αριθμό χωρίς το τελευταίο του ψηφίο.

Ώρα για άσκηση:

Άσκηση 1.4
Ποιο είναι το αποτέλεσμα του επόμενου προγράμματος στην οθόνη;

```
#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
  int a,b,c,d;
  a=1253;
  b=a%10;
  a=a/10;
  c=a % 10;
  a=a/10;
  d=a%10;
  a=a/10;
  cout<<a<<" "<<b<<" "<<c<<" "<<d<<" "<<a+b+c+d<<endl;
  return 0;
}
```

Προτεραιότητα Τελεστών

Η σειρά με την οποία ενεργούν οι τελεστές είναι:

( )	παρενθέσεις
*, /, %	πολλαπλασιασμός, διαίρεση, υπόλοιπο ακέραιας διαίρεσης
+, -	πρόσθεση, αφαίρεση

Προσαρμογή τύπου – Casting.

Υπάρχουν περιπτώσεις κατά τις οποίες πρέπει να μετατρέψουμε δεδομένα από ένα τύπο σε άλλο. Κλασική περίπτωση είναι η διαίρεση μεταξύ δύο ακεραίων τιμών, όπου το αποτέλεσμα θα είναι το ακέραιο μέρος της διαίρεσης.

Για παράδειγμα το 10/4 θα δώσει απάντηση 2 γιατί το 4 πάει μόνο δύο φορές στο 10. Αν θέλω να έχω κανονική διαίρεση, τότε πρέπει να μετατρέψω τον ένα από τους δύο αριθμούς σε τύπο **float**. Αυτό γίνεται βάζοντας το **(float)** μπροστά από την τιμή ή την μεταβλητή.

Το επόμενο πρόγραμμα παρουσιάζει τον τρόπο:

Παράδειγμα

```
#include <iostream>
using namespace std;
int main()
{
  int a,b,c; float r;
  a=10;
```

```
  b=4;
  c=a/b;
  cout<<c<<endl;  //θα παρουσιάσει 2
  r=(float)a /b;
  cout<<r<<endl;  //θα παρουσιάσει 2.5
  return 0;
}
```

Βιβλιοθήκη cmath.

Παρακάτω εμφανίζονται κάποιες χρήσιμες συναρτήσεις που υπάρχουν στη βιβλιοθήκη <cmath>:

sqrt(x)	**τετραγωνική ρίζα**
abs(x)	**απόλυτη τιμή**
pow(x,y)	**αποτέλεσμα της δύναμης x^y**
trunc(x)	**επιστρέφει το ακέραιο μέρος του αριθμού x**
round(x)	**επιστρέφει τον αριθμό x στρογγυλοποιημένο**

Παράδειγμα:

```
#include<iostream>
#include<cmath>       // περιλαμβάνει τις πιο κάτω συναρτήσεις
using namespace std;

int main() {
 cout << sqrt(144.0) << endl;              // 12
 cout << pow(5.0, 2.0) << endl;            // 25
 cout << abs(-10.0) << endl;               // 10
 cout << trunc(5.9) << endl;               // 5
 cout << round(5.9) << endl;               // 6
return 0;
}
```

Λύσεις ασκήσεων:

Άσκηση 1.1

```
#include <iostream>
using namespace std;
int main()
{
  cout<<"*"<<endl;
  cout<<"**"<<endl;
  cout<<"***"<<endl;
  cout<<"****"<<endl;
  cout<<"*****"<<endl;
  cout<<"******"<<endl;
  return 0;
}
```

Άσκηση 1.2

□□This□is□a□cat
□1144222
3.142

Άσκηση 1.3

```
#include <iostream>
using namespace std;
int main()
{  int sum, mult;
   int a,b,c;
   cin>>a>>b>>c;
   sum=a+b+c;
   mult=a*b*c;
   cout<<sum<<" "<<mult;
  return 0;
}
```

Άσκηση 1.4

```
1 3 5 2 11
```