

Streams

Input / Output in C++

George Kastrinis
(gkastrinis@di.uoa.gr)

Streams – What?

- Η είσοδος και η έξοδος δεδομένων, στη C++ βασίζεται στην έννοια των “ρευμάτων” - *streams*
- Ένα ρεύμα είναι ουσιαστικά μια ακολουθία χαρακτήρων (γενικότερα bytes)
- Ένα ρεύμα μπορεί να είναι για είσοδο, για έξοδο ή και για τα δύο

Streams – What?

- Η είσοδος και η έξοδος δεδομένων, στη C++ βασίζεται στην έννοια των “ρευμάτων” - *streams*
- Ένα ρεύμα είναι ουσιαστικά μια ακολουθία χαρακτήρων (γενικότερα bytes)
- Ένα ρεύμα μπορεί να είναι για είσοδο, για έξοδο ή και για τα δύο

Streams – What?

- Η είσοδος και η έξοδος δεδομένων, στη C++ βασίζεται στην έννοια των “ρευμάτων” - *streams*
- Ένα ρεύμα είναι ουσιαστικά μια ακολουθία χαρακτήρων (γενικότερα bytes)
- Ένα ρεύμα μπορεί να είναι για είσοδο, για έξοδο ή και για τα δύο

Streams – Why?

- Η έννοια του ρεύματος ταιριάζει αρκετά με την αντικειμενοστρεφή προσέγγιση
- Ένα επίπεδο αφάιρησης
- Η είσοδος και η έξοδος αντιμετωπίζονται σαν οντότητες από τις οποίες διαβάζουμε ή γράφουμε δεδομένα

Streams – Why?

- Η έννοια του ρεύματος ταιριάζει αρκετά με την αντικειμενοστρεφή προσέγγιση
- Ένα επίπεδο αφαίρεσης
- Η είσοδος και η έξοδος αντιμετωπίζονται σαν οντότητες από τις οποίες διαβάζουμε ή γράφουμε δεδομένα

Streams – Why?

- Η έννοια του ρεύματος ταιριάζει αρκετά με την αντικειμενοστρεφή προσέγγιση
- Ένα επίπεδο αφαίρεσης
- Η είσοδος και η έξοδος αντιμετωπίζονται σαν οντότητες από τις οποίες διαβάζουμε ή γράφουμε δεδομένα

Streams – How?

- Η C++ προσφέρει ρεύματα συσχετισμένα με:
- Την standard είσοδο / έξοδο
- Αρχεία
- Συμβολοσειρές

Streams – How?

- Η C++ προσφέρει ρεύματα συσχετισμένα με:
- Την standard είσοδο / έξοδο
- Αρχεία
- Συμβολοσειρές

Streams – How?

- Η C++ προσφέρει ρεύματα συσχετισμένα με:
- Την standard είσοδο / έξοδο
- Αρχεία
- Συμβολοσειρές

Streams – How?

- Η C++ προσφέρει ρεύματα συσχετισμένα με:
- Την standard είσοδο / έξοδο
- Αρχεία
- Συμβολοσειρές

Οθόνη & Πληκτρολόγιο

- Ορισμένα από την αρχή του προγράμματος, στο namespace `std` υπάρχουν τα *cin*, *cout*, *cerr*
- Στιγμιότυπα των κλάσεων *istream* και *ostream*
- Include το *iostream* header

Οθόνη & Πληκτρολόγιο

- Ορισμένα από την αρχή του προγράμματος, στο namespace `std` υπάρχουν τα *cin*, *cout*, *cerr*
- Στιγμιότυπα των κλάσεων *istream* και *ostream*
- Include το *iostream* header

Οθόνη & Πληκτρολόγιο

- Ορισμένα από την αρχή του προγράμματος, στο namespace `std` υπάρχουν τα *cin*, *cout*, *cerr*
- Στιγμιότυπα των κλάσεων *istream* και *ostream*
- Include το *iostream* header

Οθόνη

- Τα ρεύματα cout και cerr είναι συνδεδεμένα με την standard έξοδο
- Το cout προορίζεται για την “κανονική” εκτύπωση μηνυμάτων του προγράμματος
- Το cerr προορίζεται για την εκτύπωση μηνυμάτων λαθών

Οθόνη

- Τα ρεύματα cout και cerr είναι συνδεδεμένα με την standard έξοδο
- Το cout προορίζεται για την “κανονική” εκτύπωση μηνυμάτων του προγράμματος
- Το cerr προορίζεται για την εκτύπωση μηνυμάτων λαθών

Οθόνη

- Τα ρεύματα cout και cerr είναι συνδεδεμένα με την standard έξοδο
- Το cout προορίζεται για την “κανονική” εκτύπωση μηνυμάτων του προγράμματος
- Το cerr προορίζεται για την εκτύπωση μηνυμάτων λαθών

Πληκτρολόγιο

- Το c_{in} είναι το ρεύμα που είναι συνδεδεμένο με την standard είσοδο

Παράδειγμα



ΙΟ από αρχεία

- Για είσοδο / έξοδο από αρχεία, παρέχονται τα ρεύματα:
- Για είσοδο – *ifstream* (input file stream)
- Για έξοδο – *ofstream* (output file stream)
- Και για τα δύο – *fstream* (file stream)
- Include το *fstream* header

ΙΟ από αρχεία

- Για είσοδο / έξοδο από αρχεία, παρέχονται τα ρεύματα:
- Για είσοδο – *ifstream* (input file stream)
- Για έξοδο – *ofstream* (output file stream)
- Και για τα δύο – *fstream* (file stream)
- Include το *fstream* header

ΙΟ από αρχεία

- Για είσοδο / έξοδο από αρχεία, παρέχονται τα ρεύματα:
- Για είσοδο – *ifstream* (input file stream)
- Για έξοδο – *ofstream* (output file stream)
- Και για τα δύο – *fstream* (file stream)
- Include το *fstream* header

ΙΟ από αρχεία

- Για είσοδο / έξοδο από αρχεία, παρέχονται τα ρεύματα:
- Για είσοδο – *ifstream* (input file stream)
- Για έξοδο – *ofstream* (output file stream)
- Και για τα δύο – *fstream* (file stream)
- Include το *fstream* header

ΙΟ από αρχεία

- Για είσοδο / έξοδο από αρχεία, παρέχονται τα ρεύματα:
- Για είσοδο – *ifstream* (input file stream)
- Για έξοδο – *ofstream* (output file stream)
- Και για τα δύο – *fstream* (file stream)
- Include το *fstream* header

Error flags in streams

- Τα ρεύματα έχουν 3 bits τα οποία παρέχουν πληροφορίες για την κατάσταση τους
- *eofbit* – δείχνει αν έχει τελειώσει το αρχείο με το οποίο ήταν συνδεδεμένο το ρεύμα (EOF)
- *badbit* – δείχνει αν υπάρχει πρόβλημα με το ίδιο το ρεύμα
- *failbit* – δείχνει αν υπήρξε πρόβλημα με την τελευταία ενέργεια πάνω στο ρεύμα (π.χ. διάβασμα ακεραίου, ενώ δεν υπήρχε ακέραιος για να διαβαστεί)

Error flags in streams

- Τα ρεύματα έχουν 3 bits τα οποία παρέχουν πληροφορίες για την κατάσταση τους
- *eofbit* – δείχνει αν έχει τελειώσει το αρχείο με το οποίο ήταν συνδεδεμένο το ρεύμα (EOF)
- *badbit* – δείχνει αν υπάρχει πρόβλημα με το ίδιο το ρεύμα
- *failbit* – δείχνει αν υπήρξε πρόβλημα με την τελευταία ενέργεια πάνω στο ρεύμα (π.χ. διάβασμα ακεραίου, ενώ δεν υπήρχε ακέραιος για να διαβαστεί)

Error flags in streams

- Τα ρεύματα έχουν 3 bits τα οποία παρέχουν πληροφορίες για την κατάσταση τους
- *eofbit* – δείχνει αν έχει τελειώσει το αρχείο με το οποίο ήταν συνδεδεμένο το ρεύμα (EOF)
- *badbit* – δείχνει αν υπάρχει πρόβλημα με το ίδιο το ρεύμα
- *failbit* – δείχνει αν υπήρξε πρόβλημα με την τελευταία ενέργεια πάνω στο ρεύμα (π.χ. διάβασμα ακεραίου, ενώ δεν υπήρχε ακέραιος για να διαβαστεί)

Error flags in streams

- Τα ρεύματα έχουν 3 bits τα οποία παρέχουν πληροφορίες για την κατάσταση τους
- *eofbit* – δείχνει αν έχει τελειώσει το αρχείο με το οποίο ήταν συνδεδεμένο το ρεύμα (EOF)
- *badbit* – δείχνει αν υπάρχει πρόβλημα με το ίδιο το ρεύμα
- *failbit* – δείχνει αν υπήρξε πρόβλημα με την τελευταία ενέργεια πάνω στο ρεύμα (π.χ. διάβασμα ακεραίου, ενώ δεν υπήρχε ακέραιος για να διαβαστεί)

Error flags in streams

- Για κάθε bit, υπάρχει και αντίστοιχη συνάρτηση-μέλος (*eof, fail, bad*)
- Η συνάρτηση-μέλος *good* ελέγχει αν δεν είναι set (1) κανένα από τα τρία αυτά bits
- Η συνάρτηση-μέλος *clear* μπορεί να αλλάξει τιμές στην κατάσταση του stream, και αν κληθεί χωρίς ορίσματα, κάνει unset (0) όλα τα error flags

Error flags in streams

- Για κάθε bit, υπάρχει και αντίστοιχη συνάρτηση-μέλος (*eof, fail, bad*)
- Η συνάρτηση-μέλος *good* ελέγχει αν δεν είναι set (1) κανένα από τα τρία αυτά bits
- Η συνάρτηση-μέλος *clear* μπορεί να αλλάξει τιμές στην κατάσταση του stream, και αν κληθεί χωρίς ορίσματα, κάνει unset (0) όλα τα error flags

Error flags in streams

- Για κάθε bit, υπάρχει και αντίστοιχη συνάρτηση-μέλος (*eof, fail, bad*)
- Η συνάρτηση-μέλος *good* ελέγχει αν δεν είναι set (1) κανένα από τα τρία αυτά bits
- Η συνάρτηση-μέλος *clear* μπορεί να αλλάξει τιμές στην κατάσταση του stream, και αν κληθεί χωρίς ορίσματα, κάνει unset (0) όλα τα error flags

ifstream

- Το `ifstream` μπορεί να χρησιμοποιηθεί ακριβώς όπως και το `cin`
- Με την συνάρτηση-μέλος `good`, ελέγχουμε αν η ενέργεια διαβάσματος ήταν επιτυχής
- Σε αντιστοιχία με την C, υπάρχει συνάρτηση-μέλος `eof`, που μας ενημερώνει για το τέλος του αρχείου
- Δεν είναι αναγκαστικό να κάνουμε `close` τα ρεύματα των αρχείων, καθώς κάτι τέτοιο γίνεται κατά την καταστροφή τους

ifstream

- Το `ifstream` μπορεί να χρησιμοποιηθεί ακριβώς όπως και το `cin`
- Με την συνάρτηση-μέλος `good`, ελέγχουμε αν η ενέργεια διαβάσματος ήταν επιτυχής
- Σε αντιστοιχία με την C, υπάρχει συνάρτηση-μέλος `eof`, που μας ενημερώνει για το τέλος του αρχείου
- Δεν είναι αναγκαστικό να κάνουμε `close` τα ρεύματα των αρχείων, καθώς κάτι τέτοιο γίνεται κατά την καταστροφή τους

ifstream

- Το `ifstream` μπορεί να χρησιμοποιηθεί ακριβώς όπως και το `cin`
- Με την συνάρτηση-μέλος `good`, ελέγχουμε αν η ενέργεια διαβάσματος ήταν επιτυχής
- Σε αντιστοιχία με την C, υπάρχει συνάρτηση-μέλος `eof`, που μας ενημερώνει για το τέλος του αρχείου
- Δεν είναι αναγκαστικό να κάνουμε `close` τα ρεύματα των αρχείων, καθώς κάτι τέτοιο γίνεται κατά την καταστροφή τους

ifstream

- Το `ifstream` μπορεί να χρησιμοποιηθεί ακριβώς όπως και το `cin`
- Με την συνάρτηση-μέλος `good`, ελέγχουμε αν η ενέργεια διαβάσματος ήταν επιτυχής
- Σε αντιστοιχία με την C, υπάρχει συνάρτηση-μέλος `eof`, που μας ενημερώνει για το τέλος του αρχείου
- Δεν είναι αναγκαστικό να κάνουμε `close` τα ρεύματα των αρχείων, καθώς κάτι τέτοιο γίνεται κατά την καταστροφή τους

ifstream

- Με την συνάρτηση-μέλος *peek* μπορούμε να δούμε τον επόμενο χαρακτήρα στο ρεύμα, χωρίς να αφαιρεθεί από αυτό
- Με την συνάρτηση-μέλος *ignore* μπορούμε να παραβλέψουμε ένα πλήθος χαρακτήρων από το ρεύμα
- Με την συνάρτηση-μέλος *getline* μπορούμε να διαβάσουμε μια γραμμή από το ρεύμα και να την αποθηκεύσουμε σε κάποιον πίνακα χαρακτήρων

ifstream

- Με την συνάρτηση-μέλος *peek* μπορούμε να δούμε τον επόμενο χαρακτήρα στο ρεύμα, χωρίς να αφαιρεθεί από αυτό
- Με την συνάρτηση-μέλος *ignore* μπορούμε να παραβλέψουμε ένα πλήθος χαρακτήρων από το ρεύμα
- Με την συνάρτηση-μέλος *getline* μπορούμε να διαβάσουμε μια γραμμή από το ρεύμα και να την αποθηκεύσουμε σε κάποιον πίνακα χαρακτήρων

ifstream

- Με την συνάρτηση-μέλος *peek* μπορούμε να δούμε τον επόμενο χαρακτήρα στο ρεύμα, χωρίς να αφαιρεθεί από αυτό
- Με την συνάρτηση-μέλος *ignore* μπορούμε να παραβλέψουμε ένα πλήθος χαρακτήρων από το ρεύμα
- Με την συνάρτηση-μέλος *getline* μπορούμε να διαβάσουμε μια γραμμή από το ρεύμα και να την αποθηκεύσουμε σε κάποιον πίνακα χαρακτήρων

ifstream

- Η συνάρτηση-μέλος `getline` μπορεί να αποθηκεύσει το αποτέλεσμα μόνο σε πίνακα χαρακτήρων
- `char title[256]; in.getline (title,256);`
- Για να μπορούμε να χρησιμοποιήσουμε `strings`, υπάρχει η συνάρτηση `getline` ορισμένη στο `<string>`
- `string title; getline(in, title);`

ifstream

- Η συνάρτηση-μέλος `getline` μπορεί να αποθηκεύσει το αποτέλεσμα μόνο σε πίνακα χαρακτήρων
- `char title[256]; in.getline (title,256);`
- Για να μπορούμε να χρησιμοποιήσουμε `strings`, υπάρχει η συνάρτηση `getline` ορισμένη στο `<string>`
- `string title; getline(in, title);`

ifstream & μετακίνηση στο αρχείο

- Μαζί με κάθε `input stream`, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *get pointer*
- Με την συνάρτηση-μέλος *tellg* μαθαίνουμε για το που βρίσκεται ο *get pointer*
- Με την συνάρτηση-μέλος *seekg* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

ifstream & μετακίνηση στο αρχείο

- Μαζί με κάθε input stream, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *get pointer*
- Με την συνάρτηση-μέλος *tellg* μαθαίνουμε για το που βρίσκεται ο get pointer
- Με την συνάρτηση-μέλος *seekg* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

ifstream & μετακίνηση στο αρχείο

- Μαζί με κάθε input stream, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *get pointer*
- Με την συνάρτηση-μέλος *tellg* μαθαίνουμε για το που βρίσκεται ο get pointer
- Με την συνάρτηση-μέλος *seekg* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

ifstream & μετακίνηση στο αρχείο

- Μαζί με κάθε input stream, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *get pointer*
- Με την συνάρτηση-μέλος *tellg* μαθαίνουμε για το που βρίσκεται ο get pointer
- Με την συνάρτηση-μέλος *seekg* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

Παράδειγμα ifstream



ofstream

- Το `ofstream` μπορεί να χρησιμοποιηθεί ακριβώς όπως και το `cout`
- Τα δεδομένα δεν είναι αναγκαστικό να γραφτούν αμέσως στο αρχείο (buffered IO)
- Υπάρχει η συνάρτηση-μέλος `flush()` που γράφει τα περιεχόμενα του `buffer` στο αρχείο, αμέσως μετά την κλήση της
- Κατά το κλείσιμο του αρχείου, γίνεται και γράψιμο του `buffer` στο αρχείο

ofstream

- Το ofstream μπορεί να χρησιμοποιηθεί ακριβώς όπως και το cout
- Τα δεδομένα δεν είναι αναγκαστικό να γραφτούν αμέσως στο αρχείο (buffered IO)
- Υπάρχει η συνάρτηση-μέλος *flush()* που γράφει τα περιεχόμενα του buffer στο αρχείο, αμέσως μετά την κλήση της
- Κατά το κλείσιμο του αρχείου, γίνεται και γράψιμο του buffer στο αρχείο

ofstream

- Το ofstream μπορεί να χρησιμοποιηθεί ακριβώς όπως και το cout
- Τα δεδομένα δεν είναι αναγκαστικό να γραφτούν αμέσως στο αρχείο (buffered IO)
- Υπάρχει η συνάρτηση-μέλος *flush()* που γράφει τα περιεχόμενα του buffer στο αρχείο, αμέσως μετά την κλήση της
- Κατά το κλείσιμο του αρχείου, γίνεται και γράψιμο του buffer στο αρχείο

ofstream

- Το ofstream μπορεί να χρησιμοποιηθεί ακριβώς όπως και το cout
- Τα δεδομένα δεν είναι αναγκαστικό να γραφτούν αμέσως στο αρχείο (buffered IO)
- Υπάρχει η συνάρτηση-μέλος *flush()* που γράφει τα περιεχόμενα του buffer στο αρχείο, αμέσως μετά την κλήση της
- Κατά το κλείσιμο του αρχείου, γίνεται και γράψιμο του buffer στο αρχείο

ofstream & μετακίνηση στο αρχείο

- Μαζί με κάθε `output stream`, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *put pointer*
- Με την συνάρτηση-μέλος *tellp* μαθαίνουμε για το που βρίσκεται ο *put pointer*
- Με την συνάρτηση-μέλος *seekp* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

ofstream & μετακίνηση στο αρχείο

- Μαζί με κάθε output stream, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *put pointer*
- Με την συνάρτηση-μέλος *tellp* μαθαίνουμε για το που βρίσκεται ο put pointer
- Με την συνάρτηση-μέλος *seekp* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

ofstream & μετακίνηση στο αρχείο

- Μαζί με κάθε `ofstream`, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *put pointer*
- Με την συνάρτηση-μέλος *tellp* μαθαίνουμε για το που βρίσκεται ο *put pointer*
- Με την συνάρτηση-μέλος *seekp* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

ofstream & μετακίνηση στο αρχείο

- Μαζί με κάθε `output stream`, υπάρχει και ένας δείκτης που δείχνει σε ποιο σημείο στο ρεύμα βρισκόμαστε
- Ο δείκτης είναι γνωστός και ως *put pointer*
- Με την συνάρτηση-μέλος *tellp* μαθαίνουμε για το που βρίσκεται ο *put pointer*
- Με την συνάρτηση-μέλος *seekp* μπορούμε να αλλάξουμε θέση στο δείκτη αυτό (σε αντιστοιχία με την συνάρτηση *fseek* στην C)

Παράδειγμα ofstream



Παράδειγμα fstream



Παράδειγμα `fstream`

- Το προηγούμενο παράδειγμα δεν θα δουλέψει αν το αρχείο `data3.txt` δεν υπάρχει ήδη
- Για να γίνει αυτό, πρέπει πρώτα να ανοίξουμε το αρχείο για γράψιμο ώστε να δημιουργηθεί και στη συνέχεια να το ανοίξουμε για διάβασμα και γράψιμο

IO & binary files

- Με τα `ifstream` και `ofstream` μπορούμε να χειριστούμε και δυαδικά αρχεία (αρχεία που έχουν bytes και όχι κείμενο)
- Η συνάρτηση-μέλος `read` του `ifstream`, διαβάζει από το αρχείο ένα πλήθος από bytes και το αποθηκεύει σε έναν πίνακα χαρακτήρων
- Η συνάρτηση-μέλος `write` του `ofstream`, γράφει στο αρχείο ένα πλήθος από bytes, που τα διαβάζει από έναν πίνακα χαρακτήρων

IO & binary files

- Με τα `ifstream` και `ofstream` μπορούμε να χειριστούμε και δυαδικά αρχεία (αρχεία που έχουν bytes και όχι κείμενο)
- Η συνάρτηση-μέλος `read` του `ifstream`, διαβάζει από το αρχείο ένα πλήθος από bytes και το αποθηκεύει σε έναν πίνακα χαρακτήρων
- Η συνάρτηση-μέλος `write` του `ofstream`, γράφει στο αρχείο ένα πλήθος από bytes, που τα διαβάζει από έναν πίνακα χαρακτήρων

IO & binary files

- Με τα `ifstream` και `ofstream` μπορούμε να χειριστούμε και δυαδικά αρχεία (αρχεία που έχουν bytes και όχι κείμενο)
- Η συνάρτηση-μέλος `read` του `ifstream`, διαβάζει από το αρχείο ένα πλήθος από bytes και το αποθηκεύει σε έναν πίνακα χαρακτήρων
- Η συνάρτηση-μέλος `write` του `ofstream`, γράφει στο αρχείο ένα πλήθος από bytes, που τα διαβάζει από έναν πίνακα χαρακτήρων

IO Streams & Strings



IO Streams & Strings

- Η C++ παρέχει το *stringstream* σαν ένα τρόπο για να μπορούμε να χειριστούμε ένα *string* σαν να ήταν ρευμα
- Ένα ρεύμα συνδέεται με ένα *string* και στη συνέχεια μας παρέχεται η λειτουργικότητα που θα περιμέναμε από ένα οποιοδήποτε *stream*
- Include το *sstream* header

IO Streams & Strings

- Η C++ παρέχει το *stringstream* σαν ένα τρόπο για να μπορούμε να χειριστούμε ένα `string` σαν να ήταν ρευμα
- Ένα ρεύμα συνδέεται με ένα `string` και στη συνέχεια μας παρέχεται η λειτουργικότητα που θα περιμέναμε από ένα οποιοδήποτε `stream`
- Include το *sstream* header

IO Streams & Strings

- Η C++ παρέχει το *stringstream* σαν ένα τρόπο για να μπορούμε να χειριστούμε ένα `string` σαν να ήταν ρευμα
- Ένα ρεύμα συνδέεται με ένα `string` και στη συνέχεια μας παρέχεται η λειτουργικότητα που θα περιμέναμε από ένα οποιοδήποτε `stream`
- Include το *sstream* header

istream

- Με το *istream* μπορούμε να χειριστούμε ένα `string` σαν να ήταν ρεύμα εισόδου
- Ακολουθεί ενδεικτικό παράδειγμα χρήσης, για το διάβασμα ενός ακεραίου από ένα `string`
- Αντίστοιχο της συνάρτησης `atoi`

istream

- Με το *istream* μπορούμε να χειριστούμε ένα `string` σαν να ήταν ρεύμα εισόδου
- Ακολουθεί ενδεικτικό παράδειγμα χρήσης, για το διάβασμα ενός ακεραίου από ένα `string`
- Αντίστοιχο της συνάρτησης `atoi`

ostream

- Με το *ostream* μπορούμε να χειριστούμε ένα `string` σαν να ήταν ρεύμα εξόδου
- Ακολουθεί ενδεικτικό παράδειγμα χρήσης, για το γράψιμο αριθμών σε `string`
- Αντίστοιχο της συνάρτησης `itoa`

ostream

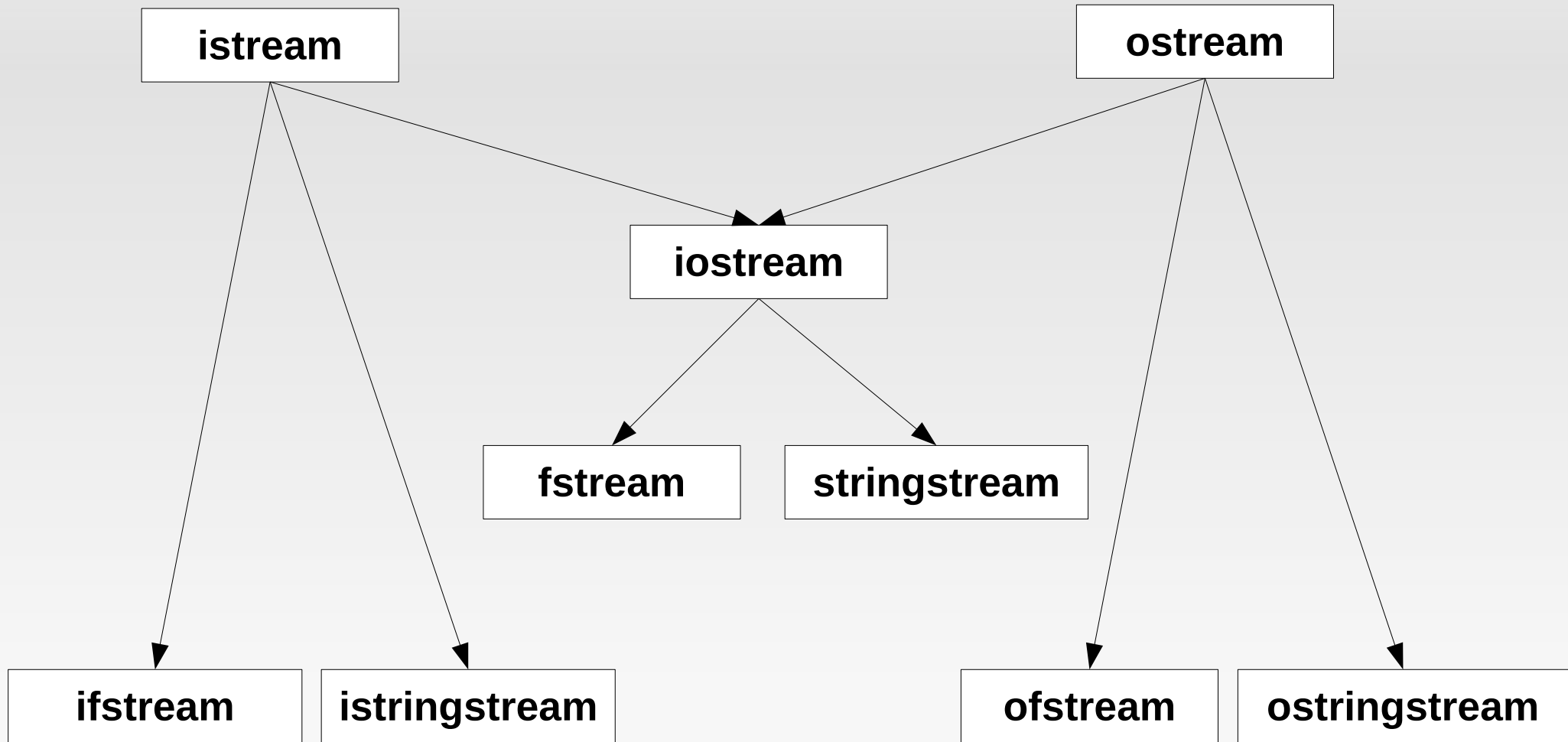
- Με το *ostream* μπορούμε να χειριστούμε ένα string σαν να ήταν ρεύμα εξόδου
- Ακολουθεί ενδεικτικό παράδειγμα χρήσης, για το γράψιμο αριθμών σε string
- Αντίστοιχο της συνάρτησης *itoa*

Reference – Χρήσιμοι Σύνδεσμοι

- cplusplus.com
- [ifstream](#)
- [ofstream](#)
- [fstream](#)
- [getline για string](#)
- [stringstream](#)
- [istringstream](#)
- [ostringstream](#)



Streams & Inheritance



Streams – more examples



Questions?

