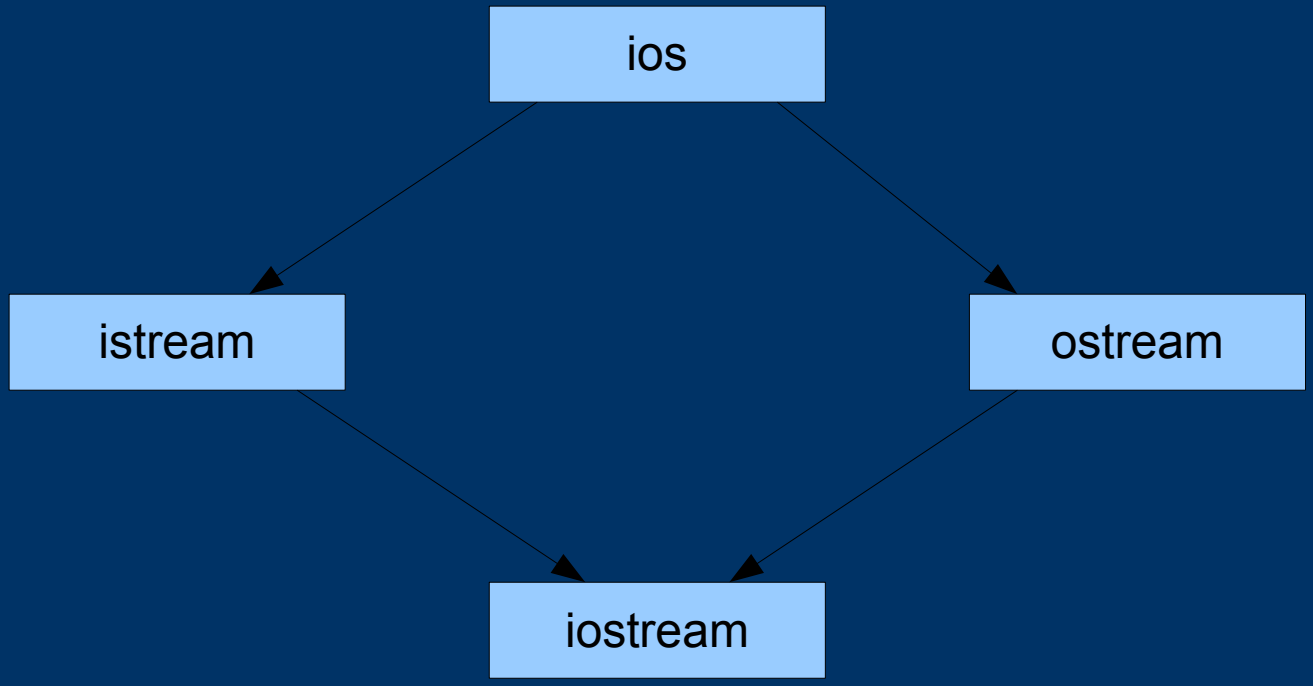
C++: Part 8

Vipin Bhatnagar

I/O Operations

- We know
  - cin>> and cout<<
  - Coming from iostream header file (included at the top of a C++ program (#include <iostream>)
  - Fall under the unformatted IO operations
- Origin
  - iostream header file where the classes are defined
  - iostream instead inherits these classes from: istream and ostream classes which inherit these from ios base class

I/O flow



Functions for IO ops:

- istream: get(), getline() and read()
- ostream: put(), write()

Member functions of istream and ostream classes
must be used with objects

```
char a;
cin.get(a);      // cin is an object predefined in iostream
```

or

```
char a;
a = cin.get();
```

Functions for I/O contd.

- For output:

```
char ans = ”Y”;
cout.put(ans);
or
cout.put('N');
```

TRY: use put() to display some ASCII values!
eg. cout.put(65); // converts *int* to a *char* & displays character A
NB: get() and put() are meant for single character

example

```
#include <iostream>
int main()
{
    int count = 0;
    char a;
    cout << “Enter Text\n”;
    cin.get(a);   // readin first char of entered text
    while(a != '\n')
    {
        cout.put(a);
        count++;
        cin.get(a);
    }
    cout << “Number of chars entered: “ << count <<”\n”;
    return (0);
}
```

getline() and write() functions

- More efficient way to readin a line of text:
  cin.getline(line,size); // usage
  Reads in the line till the newline character is seen: return
  - Newline char is read but not saved
  - Instead replaced by a null character

eg.
char name[20];
cin.getline(name,20);
OO Programming <press enter>
NB: cin>> can be used to readin BUT white spaces