First C++ Programs

C++ Standard Library
Sample programs

C++ Standard Library:

Just like C has a standard library, C++ also has one (you can look them up).

Two good resources:
https://www.tutorialspoint.com/cpp_standard_library
https://msdn.microsoft.com/en-us/library/cscc687y.aspx

Sample Programs:

Notes:

1. **DO NOT CODE EVERYTHING IN MAIN!!! This will be an automatic 0 for any assignment, quiz or exam.**
   - For the next couple of classes, we will be using functions
   - In future lectures, we will be using classes

2. **Do not forget to indent**
   - It is impossible to read and understand your code if you do not indent

3. **Make sure any comments you include are meaningful**
   - Code overall idea, not every single line

Program 1:
Emee wants to create a program that adds friends to a group. There should be no duplicate friends.

Notice our compiler is g++ (in 1320, we used gcc for C)

```
computer$ g++ friends.cpp
computer$ ./a.out
Enter the max number of friends for the group:
3
Enter friend 1:
Bob
Enter friend 2:
Jon
Enter friend 3:
Bob
Sorry, there is already a friend with this name.
Enter friend 3:
Carl
All friends added! :)
```

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

//this function returns true if word is there, false otherwise
bool check_duplicate(vector <string> s, string word)
{
   bool ret=false;

   for(int i=0;i<s.size()&& !ret;i++)
   {
     if(s[i]==word) //found it-duplicate
     {
       ret=true;
     }
   }

   return ret;
}

int main(int argc, char **argv)
{
  int max_num;
  int i=0;
  string name;
  vector <string> all_friends;

  cout<<"Enter the max number of friends for the group: "<<endl;
  cin>>max_num;


  //Allow users to enter friends until max num is reached
  while(i<max_num)
  {
    cout<<"Enter friend "<<(i+1)<<":"<<endl;
    cin>>name;

    if(!check_duplicate(all_friends,name)) //Go ahead and add if not a duplicate
    {
      all_friends.push_back(name);
      i++;
    }

    else //Otherwise don't add
    {
      cout<<"Sorry, there is already a friend with this name."<<endl;
    }
```

```
  }

  cout<<"All friends added! :)"<<endl;
}
```

Variation 1 (using *stringstream*-we will discuss what that means in a later class):

```
computer$ g++ friends_variation1.cpp
computer$ ./a.out
Enter all 3 friends for the group:
Jon Jane Jill
Jon
Jane
Jill
All friends added! :)
```

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


int main(int argc, char **argv)
{
  string all_friends;

  cout<<"Enter all 3 friends for the group: "<<endl;
  getline(cin,all_friends);
  stringstream ss(all_friends);

  string friend1;
  string friend2;
  string friend3;

  ss>>friend1>>friend2>>friend3; //separated out all names by space (this method only works when you know how many words will be separated by space

  cout<<friend1<<endl;
  cout<<friend2<<endl;
  cout<<friend3<<endl;

  cout<<"All friends added! :)"<<endl;
}
```

```
computer$ g++ friends_variation2.cpp
computer$ ./a.out
```

```
Enter friends:
Jon Jill Will Bill
Total friends: 4
Jon
Jill
Will
Bill
```

Variation 2:

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



int main(int argc, char **argv)
{
  string all_friends;
  string single_name;
  vector<string> friend_vector;

  cout<<"Enter friends: "<<endl;
  getline(cin,all_friends);
  stringstream ss(all_friends);

  //there are other ways to do this, this is just one way.  We are basically adding every string separated by space to a vector.
  while(ss>>single_name)
  {
    friend_vector.push_back(single_name);
  }

  cout<<"Total friends: "<<friend_vector.size()<<endl;

  for(int i=0;i<friend_vector.size();i++)
  {
    cout<<friend_vector.at(i)<<endl; //you can use [] instead of the at() function
  }

}
```

Program 2:

Create a program that allows you to add money to your wallet. The program should keep adding money until it hits a specified goal.

```
computer$ g++ -o wallet wallet.cpp
computer$ ./wallet
Enter your goal amount:
5
Nothing in your wallet yet.
Enter amount to add to wallet:
2
Enter amount to add to wallet:
3
You hit your goal without going over.
```

```
#include <iostream>
#include <vector>
#include <string>

//not using namespace std;

//Returns the total value in the wallet
int wallet_total(std::vector <int> money)
{
   if(money.size()==0) //nothing in wallet yet
   {
     std::cout<<"Nothing in your wallet yet."<<std::endl;
     return 0;
   }

   int total=0;

   for(int i=0;i<money.size();i++)
   {
     total+=money[i];
   }

   return total;
}

//Checks the total against the goal
bool check_total(int total, int goal)
{

  bool b;
  if(total<=goal)
  {
    std::cout<<"You hit your goal without going over."<<std::endl;
    b= true;
  }
```

```
  else
  {
    std::cout<<"You went over your goal."<<std::endl;
    b= false;
  }

  return b;
}


int main(int argc, char **argv)
{
  int goal;
  int add;
  std::vector <int> wallet;

  std::cout<<"Enter your goal amount:"<<std::endl;
  std::cin>>goal;

  while(wallet_total(wallet)<goal)
  {
    std::cout<<"Enter amount to add to wallet: "<<std::endl;
    std::cin>>add;
    wallet.push_back(add);
  }

  //Notice I'm using a function here with a return type but not using the return type
  check_total(wallet_total(wallet),goal);
}
```