

# CSCI 127: Introduction to Computer Science



[hunter.cuny.edu/csci](https://hunter.cuny.edu/csci)

# Announcements

- **Final Exam is on MONDAY MAY 20 at 9:00 AM**
- Room 118 Hunter North (Assembly Hall), ground floor of the North Building
- **Seating assignment will be available on Blackboard/My Grades this week**
- **Next Tuesday, May 14, we will have a Mock Exam**
  - ▶ Only 1 hr 15 mins for the Mock, 2 hours for the real exam.
  - ▶ Just a practice run and it will NOT be graded (answer keys will be posted).
  - ▶ However, the mock will have the same logistics and question format as the real final exam.

# Announcements

- If you can't make the scheduled exam time you can take the conflict exam
- **Wednesday, May 15 at 9 AM in room 1001G HN**
- Please email me (melissa.lynch@hunter.cuny.edu) to sign up for the conflict exam
- Important! If you take the early exam you will NOT be able to take the regular exam on 5/20

# Frequently Asked Questions

- What's the best way to study for the final exam?  
*The final exam problems are variations on the homework, quizzes, lecture examples, and lecture previews.*  
*Past exams (and answer keys) are online. Do 7-10 previous exams: allow 1 hour (half time) and work through, grade yourself, update your note sheet, and repeat.*
- I'm worried about my grade. Should I do Pass/NoCredit?  
*It's fine with us, but check with your advisor to make sure it's accepted for your program of study.*
- Why do you care about cheating?  
*First: it gives unfair advantage & is immoral.*  
*Second: it degrades the quality of our students.*  
*Third: it's a standard question on faculty references.*  
*Industry & graduate schools hate it: don't want someone who falsifies work.*

# Today's Topics

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout << endl << "Lbs: " << lbs << "\n\n";
    return 0;
}
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- Indefinite Loops in C++

# Today's Topics

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout << endl << "Lbs: " << lbs << "\n\n";
    return 0;
}
```

- **Recap: I/O & Definite Loops in C++**
- Conditionals in C++
- Indefinite Loops in C++

# Recap: Basic Form & I/O in C++

---

```
//C++ program demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout << endl << "Lbs: " << lbs << "\n\n";
    return 0;
}
```

---

# Recap: Definite Loops in C++

---

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

int main() {
    int lines;
    cout << "Enter a number: ";
    cin >> lines;
    for(int i = 1; i <= lines; i++) {
        for(int j = 0; j < i; j++){
            cout << "*";
        }
        cout << "\n";
    }
}
```

---



# Recap: Basic Form & I/O in C++

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout << endl << "Lbs: " << lbs << "\n\n";
    return 0;
}
```

- Efficient for systems programming.
- Programs are organized in functions.
- Must declare variables: `int num;`
- Many types available:  
`int, float, char, ...`
- To print: `cout << "Hello!!";`
- To get input: `cin >> num;`
- To use those I/O functions:  
`#include <iostream>`  
`using namespace std;`
- Definite loops:  
`for (i = 0; i < 10; i++) {...}`
- Blocks of code uses '{' and '}'.
- Commands generally end in ';'.

# Today's Topics

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout << endl << "Lbs: " << lbs << "\n\n";
    return 0;
}
```

- Recap: I/O & Definite Loops in C++
- **Conditionals in C++**
- Indefinite Loops in C++

# Challenge:

Predict what the following pieces of code will do:

```
//Demonstrates conditionals
#include <iostream>
using namespace std;

int main ()
{
    int yearBorn;
    cout << "Enter year born: ";
    cin >> yearBorn;
    if (yearBorn < 1946)
    {
        cout << "Greatest Generation";
    }
    else if (yearBorn <= 1964)
    {
        cout << "Baby Boomer";
    }
    else if (yearBorn <= 1984)
    {
        cout << "Generation X";
    }
    else if (yearBorn <= 2004)
    {
        cout << "Millennial";
    }
    else
    {
        cout << "TBD";
    }
}
```

return 0:  
CSCI 127 (Hunter)

```
using namespace std;

int main ()
{
    string conditions = "blowing snow";
    int winds = 100;
    float visibility = 0.2;

    if ( ( (winds > 35) && (visibility < 0.25) ) ||
        ( (conditions == "blowing snow") ||
          (conditions == "heavy snow") ) )
        cout << "Blizzard!\n";

    string origin = "South Pacific";

    if (winds > 74)
        cout << "Major storm, called a ";
    if ((origin == "Indian Ocean")
        || (origin == "South Pacific"))
        cout << "cyclone.\n";
    else if (origin == "North Pacific")
        cout << "typhoon.\n";
    else
        cout << "hurricane.\n";
}
```

# Conditionals

General format:

```
//Demonstrates conditionals
#include <iostream>
using namespace std;

int main ()
{
    int yearBorn;
    cout << "Enter year born: ";
    cin >> yearBorn;
    if (yearBorn < 1946)
    {
        cout << "Greatest Generation";
    }
    else if (yearBorn <= 1964)
    {
        cout << "Baby Boomer";
    }
    else if (yearBorn <= 1984)
    {
        cout << "Generation X";
    }
    else if (yearBorn <= 2004)
    {
        cout << "Millennial";
    }
    else
    {
        cout << "TBD";
    }
    return 0;
}
```

```
if ( logical expression )
{
    command1;
    ...
}
else if ( logical expression )
{
    command1;
    ...
}
else
{
    command1;
    ...
}
```

# Logical Operators in C++

Very similar, just different names: `&&`, `||`, and `!`:

## and (`&&`)

in1		in2	<i>returns:</i>
False	<code>&amp;&amp;</code>	False	False
False	<code>&amp;&amp;</code>	True	False
True	<code>&amp;&amp;</code>	False	False
True	<code>&amp;&amp;</code>	True	True

## or (`||`)

in1		in2	<i>returns:</i>
False	<code>  </code>	False	False
False	<code>  </code>	True	True
True	<code>  </code>	False	True
True	<code>  </code>	True	True

## not (`!`)

	in1	<i>returns:</i>
<code>!</code>	False	True
<code>!</code>	True	False

# Lecture Slip

- Write a complete C++ program that prompts the user to enter a time (in 24-hour format) and prints the time of day: morning, afternoon, or evening.
- Assume that afternoon is any time after 12 P.M. (1200), and that the evening is any time after 6 P.M. (1800).

# Lecture slip

---

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

int main() {
    int time;
    cout << "Enter time in 24 hour format: ";
    cin >> time;

    if (time < 1200) {
        cout << "Morning";
    } else if (time > 1800) {
        cout << "Evening";
    } else {
        cout << "Afternoon";
    }
    cout << "\n";
}
```

# Today's Topics

```
//Another C++ program, demonstrating I/O & arithmetic
#include <iostream>
using namespace std;

int main ()
{
    float kg, lbs;
    cout << "Enter kg: ";
    cin >> kg;
    lbs = kg * 2.2;
    cout << endl << "Lbs: " << lbs << "\n\n";
    return 0;
}
```

- Recap: I/O & Definite Loops in C++
- Conditionals in C++
- **Indefinite Loops in C++**



# Challenge: predict what the code will do

---

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

int main ()
{
    int population = 100;
    int year = 0;
    cout << "Year\tPopulation\n";
    while (population < 1000)
    {
        cout << year << "\t" << population << "\n";
        population = population * 2;
        year++;
    }
    return 0;
}
```

---

# C++ Demo

```
///While Growth Example
#include <iostream>
using namespace std;

int main ()
{
    int population = 100;
    int year = 0;
    cout << "Year\tPopulation\n";
    while(population < 1000)
    {
        cout << year << "\t\t" << population << "\n";
        population = population * 2;
        year++;
    }
    return 0;
}
```

(Demo with onlinedb)

# Indefinite Loops: while

```
///While Growth Example
#include <iostream>
using namespace std;

int main ()
{
    int population = 100;
    int year = 0;
    cout << "Year\tPopulation\n";
    while(population < 1000)
    {
        cout << year << "\t\t" << population << "\n";
        population = population * 2;
        year++;
    }
    return 0;
}
```

General format:

```
while ( logical expression )
{
    command1;
    command2;
    command3;
    ...
}
```

# Challenge: predict what the code does

---

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

int main ()
{
    int num;
    cout << "Enter an even number: ";
    cin >> num;
    while (num % 2 != 0)
    {
        cout << "\nThat's odd!\n";
        cout << "Enter an even number: ";
        cin >> num;
    }
    cout << "You entered: " << num << ".\n";
    return 0;
}
```

# C++ Demo

```
//Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
    int num;
    cout << "Enter an even number: ";
    cin >> num;
    while (num % 2 != 0)
    {
        cout << "\nThat's odd!\n";
        cout << "Enter an even number: ";
        cin >> num;
    }
    cout << "You entered: "
         << num << ".\n";
    return 0;
}
```

(Demo with onlinedb)

# Indefinite Loops: while

```
//Demonstrates loops
#include <iostream>
using namespace std;

int main ()
{
    int num;
    cout << "Enter an even number: ";
    cin >> num;
    while (num % 2 != 0)
    {
        cout << "\nThat's odd!\n";
        cout << "Enter an even number: ";
        cin >> num;
    }
    cout << "You entered: "
         << num << ".\n";
    return 0;
}
```

General format:

```
while ( logical expression )
{
    command1;
    command2;
    command3;
    ...
}
```

## Challenge: predict what the code will do

---

```
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
    int num;
    do
    {
        cout << "Enter an even number: ";
        cin >> num;
    } while (num % 2 != 0);

    cout << "You entered: " << num << ".\n";
    return 0;
}
```

# C++ Demo:

```
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
    int num;
    do
    {
        cout << "Enter an even number: ";
        cin >> num;
    } while (num % 2 != 0);

    cout << "You entered: "
         << num << ".\n";
    return 0;
}
```

(Demo with onlinedb)



# Indefinite Loops: do-while

```
//Demonstrates do-while loops
#include <iostream>
using namespace std;

int main ()
{
    int num;
    do
    {
        cout << "Enter an even number: ";
        cin >> num;
    } while (num % 2 != 0);

    cout << "You entered: "
         << num << ".\n";
    return 0;
}
```

General format:

```
do
{
    command1;
    command2;
    command3;
    ...
} while ( logical expression );
```

# Lecture slip

---

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

int main() {
    int time;
    do {
        cout << "Enter time in 24 hour format: ";
        cin >> time;
    } while (time < 0 || time > 2400);

    if (time < 1200) {
        //...
    }
}
```

---

# Lecture slip

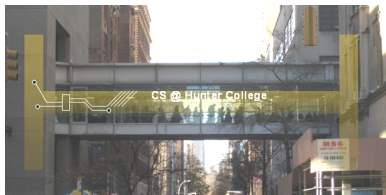
---

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

int main() {
    int numYears;
    cout << "Enter number of years: ";
    cin >> numYears;
    double p = 334.23; //initial population
    double B = 12.4/1000;
    double D = 8.4/1000;

    for(int i = 2023; i < 2023+numYears; i++) {
        //print current population
        cout << "Year\t" << i << "\t"<< p << endl;
        //calculate next year's expected population
        p = p + B*p - D*p;
    }
}
```

# Weekly Reminders!



Before the next lecture, don't forget to:

- Work on this week's Online Lab
- Schedule an appointment to take the Quiz in lab 1001G Hunter North
- Submit this week's programming assignments
- If you need help, schedule an appointment for Tutoring in lab 1001G
- Take the Lecture Preview on Blackboard on Monday (or no later than 10:15am on Tuesday)

# Lecture Slips & Writing Boards



- Hand your lecture slip to a UTA.
- Return writing boards as you leave.