

Lab 3 – Take Home Lab

If you are in the Monday lab, you can choose to go to a Tuesday-Friday lab or complete the lab on your own. If you are in a Tuesday-Friday lab, you can go to your designated lab or complete this on your own. If you choose to do this on your own, you will get checked off at the beginning of week 4 lab, and if you attend a lab, you can get check off for the work you complete as well as make up as many points as needed outside of lab.

CS 16X Research Consent

I have two ongoing research studies in CS 16X to study the impacts of different pedagogies, a design-first curriculum, and peer reviews. At this time, please let me know if you consent or not to be in these studies. No identifiable information is released, and you are not asked to do anything different in the class.

All CS 161 Students

http://oregonstate.qualtrics.com/jfe/form/SV_4V2aE9PMtl2ELHv

Only if you took CS 160

http://oregonstate.qualtrics.com/jfe/form/SV_0qyZkibC2jhPTnP

Quiz (3 pts, .5 each) Write your answers to your quiz on a separate sheet of paper to turn in to your TA during week 3 lab or at the beginning of week 4 lab, if you choose to do the lab on your own.

1. What is the late policy for assignments (describe in your own words)?
2. What should you do if you need to miss a scheduled demo?
3. What happens if you demo an assignment past two weeks from the due date, even if the assignment was submitted late?
4. If a student has a nearly-complete assignment that does not compile, what percentage should they expect on that assignment?
5. When are your peer reviews due for an assignment?
6. What happens when you submit your design late?

Worksheet (2 pts)

1. List and label the relational and logical operators, and what's the difference between them?
2. Give examples of when you use logical and relational operators in your everyday life.

3. What is short circuiting and when is it useful?
4. Can you write the following code without using any logical operators?

```
cout << "Are you hungry? (0-no, 1, yes) << endl;
cin >> choice;
if(choice == 1 && store == OPEN) {
    //get some food from the store
}
else if(choice == 1 && store == CLOSED) {
    //make some food at home
}
else {
    //I'm not hungry
}
```

Prepare for Assignment 3: Using loops

(2 pts) Design

Design is very important when developing programs and there are a variety of ways to approach it. You may draw pictures, write it out in prose or structured text, use pseudo code, and more! The point of design is to give you a blueprint to follow while you are coding. This saves time debugging your program as you can catch mistakes early. It is better to spend one hour of designing than it is to spend five hours debugging.

For this lab you must design a solution to the following problem statement. You will implement your solution.

Problem Statement

Prime numbers are important in mathematics. However, it is hard for humans to determine whether a large number is prime. You are tasked with creating a program that will determine whether a given integer is a prime number. Prime numbers are defined as any number greater than one that is only divisible by one and itself. The program should correctly handle the invalid integer inputs. In addition, user should be able to test as many integers as he or she wants in a single run. In other words, the program should not end unless the user tells you to.

For example:

Please enter a positive whole number: 1

1 is not a prime number.

Do you want to try a different value (0-no, 1-yes)? 1

Please enter a positive whole number: 19

19 is a prime number.

Do you want to try a different value (0-no, 1-yes)? 1

Please enter a positive whole number: -5

Invalid input!
Do you want to try a different value (0-no, 1-yes)? 0

(3 pts) Implementation

If you are in lab, get your design checked off by a TA before proceeding to code.

Write a program called `is_prime.cpp` to implement your solution to the problem stated above, and determine whether the following 4 integer user inputs are prime:

1139
56789
96721
86978369

You can implement a loop using a `while()` or a `for()` in C++.

The `while` loop is very much like an `if` statement, but it continues to repeat a block of code while the condition is true, instead of doing it only once.

```
cout << "Enter a number greater than 1: ";  
cin >> input;  
while(input<=1) {  
    cout << "This is not a good number!" << endl;  
    cout << "Enter a number greater than 1: ";  
    cin >> input;  
}  
//now we know we have good input greater than 1
```

The `for` loop looks very different, and it is used to repeat something a specific number of times, e.g. we want to read 4 numbers from the user to determine if they are prime numbers.

```
for(int i=0; i<4; i++) {  
    //this would be repeated 4 times  
}
```

This is what is going on with the pieces of information in the `for` loop:

Step 1: create and initialize a counter variable `i`

Step 2: check if the counter variable `i` is less than 4

Possible Step 3: do what is in the curly braces and part of the `for` loop

Possible Step 4: increment the counter variable `i` using `i++`

Possible Step 5: go back to Step 2 to see if the counter variable `i` is less than 4

Possible Step 3: do not repeat and go to the bottom of the `for` loop

Show your completed work to the TAs for credit. You will not get points if you do not get checked off!