Topics

1. Conditional statements / Loops
2. Pass by reference/ value

Selection - Basic Structure

```
if (condition1) {
        //do this if condition1 is true
}
else if (condition2) {
        //do this if condition2 is true
}
else {
        //do this if none of the above conditions are met
}
```

Iteration - Basic Structure

```
for (int i=0; i<n; i++) {
        //do this n times
}
```

```
while(condition) {
        //do this as long as condition is met
}
```

Functions - Basic Structure

```
return_type function_name(input1, input2, input3) {
        //decide output based on input
        return output_variable;
}
```

Functions - Example

```
double computeMin(double num1, double num2) {
        if (num1 < num2) {
                return num1;
        }
        else {
                return num2;
        }
}

int main() {
        double minimum = computeMin(pi,e);
        cout << "minimum is " << minimum << endl;
}
```

Define your function

Call your function in main

Agenda

- Lecture Topic Review
- **Practice Questions**
- Lab 6 assignment
- Weekly Reminders
- Q&A

Practice Problem #1

What's the output of the following program?

```
#include <iostream>
#include <string>
using namespace std;
void modifyName(string originalName) {
     string newName = "Goofus D. Dawg";
     originalName = newName;
}

int main() {
     string myName = "Donald Duck";
     modifyName(myName);
     cout << myName;
     myName = "Donald D";
     return 0;
}
```

A. Goofus D. Dawg
B. Donald Duck
C. Donald D

Practice Problem #1 Solution

What's the output of the following program?

```
#include <iostream>
#include <string>
using namespace std;
void modifyName(string originalName) {
      string newName = "Goofus D. Dawg";
      originalName = newName;
}

int main() {
      string myName = "Donald Duck";
      modifyName(myName);
      cout << myName;
      myName = "Donald D";
      return 0;
}
```

A. Goofus D. Dawg
B. Donald Duck
C. Donald D

Practice Problem #2

A

```
int age;
cout << "Please input your age: " << endl;
cin >> age;

if (age <= 14 && age >= 4) {
    if (age % 2 == 0) {
        cout << "You get 2 free meals!" << endl;
    }
    else {
        cout << "You get one free meal!" << endl;
    }
}
else {
    cout << "Sorry, no such thing as a free lunch!" << endl;
}
```

B

```
int age;
cout << "Please input your age: " << endl;
cin >> age;

if (age <= 14 && age >= 4) {
    cout << "You get one free meal!" << endl;
}
else if (age % 2 == 0) {
    cout << "You get 2 free meals!" << endl;
}
else {
    cout << "Sorry, no such thing as a free lunch!" << endl;
}
```

C

```
int age;
cout << "Please input your age: " << endl;
cin >> age;

if (age % 2 == 0) {
    cout << "You get 2 free meals!" << endl;
}
else if (age <= 14 && age >= 4) {
    cout << "You get one free meal!" << endl;
}
else {
    cout << "Sorry, no such thing as a free lunch!" << endl;
}
```

Restaurant promotion!
Individuals between ages of 4 and 14 get one meal free.

Individuals whose age is ANY even number two free meals, even if they're between 4 and 14.

Everyone else don't get a free meal.
Which option on the right is the correct implementation?

Practice Problem #2: Solution

A

```
int age;
cout << "Please input your age: " << endl;
cin >> age;

if (age <= 14 && age >= 4) {
    if (age % 2 == 0) {
        cout << "You get 2 free meals!" << endl;
    }
    else {
        cout << "You get one free meal!" << endl;
    }
}
else {
    cout << "Sorry, no such thing as a free lunch!" << endl;
}
```

B

```
int age;
cout << "Please input your age: " << endl;
cin >> age;

if (age <= 14 && age >= 4) {
    cout << "You get one free meal!" << endl;
}
else if (age % 2 == 0) {
    cout << "You get 2 free meals!" << endl;
}
else {
    cout << "Sorry, no such thing as a free lunch!" << endl;
}
```

C

```
int age;
cout << "Please input your age: " << endl;
cin >> age;

if (age % 2 == 0) {
    cout << "You get 2 free meals!" << endl;
}
else if (age <= 14 && age >= 4) {
    cout << "You get one free meal!" << endl;
}
else {
    cout << "Sorry, no such thing as a free lunch!" << endl;
}
```

Restaurant promotion!
Individuals between ages of 4 and 14 get one meal free.

Individuals whose age is ANY even number two free meals, even if they're between 4 and 14.

Everyone else don't get a free meal.
Which option on the right is the correct implementation?

Practice Problem #3

This is a classic for loop exercise.

Say we want to print a right triangle using asterisk like below:

```
* * * * *
* * * *
* * *
* *
*
```

Which option on the right is the correct implementation?

```
// Option A
for (int n = 5; n > 0; n--) {
    for (int j = 0; j < n; j++) {
        cout << "* ";
    }
    cout << endl;
}
// Option B
for (int n = 5; n > 0; n--) {
    for (int j = 0; j <= n; j++) {
        cout << "* ";
    }
    cout << endl;
}
```

```
// Option C
for (int n = 0; n < 5; n++) {
    for (int j = 0; j < n; j++) {
        cout << "* ";
    }
    cout << endl;
}
// Option D
for (int n = 0; n < 5; n++) {
    for (int j = 0; j <=  n; j++) {
        cout << "* ";
    }
    cout << endl;
}
```

Practice Problem #3 Solution

This is a classic for loop exercise.

Say we want to print a right triangle using asterisk like below:

```
* * * * *
* * * *
* * *
* *
*
```

Which option on the right is the correct implementation?

```
// Option A
for (int n = 5; n > 0; n--) {
    for (int j = 0; j < n; j++) {
        cout << "* ";
    }
    cout << endl;
}
// Option B
for (int n = 5; n > 0; n--) {
    for (int j = 0; j <= n; j++) {
        cout << "* ";
    }
    cout << endl;
}
```

```
// Option C
for (int n = 0; n < 5; n++) {
    for (int j = 0; j < n; j++) {
        cout << "* ";
    }
    cout << endl;
}
// Option D
for (int n = 0; n < 5; n++) {
    for (int j = 0; j <=  n; j++) {
        cout << "* ";
    }
    cout << endl;
}
```

Agenda

- Lecture Topic Review
- Practice Questions
- **Lab 6 assignment**
- Weekly Reminders
- Q&A

Testing tool: **`assert`** statements

```
assert(test == expected_result);
```

- Assert statements check if a statement is true
- Use them to test if behavior meets your expectations
- If the test passes → does nothing
- If test fails → throws an informative error

Assert Statement Example

```
#include <cassert>
  …
assert(computeMinimum(2, 3) == 2);
assert(computeMinimum(2, 3) == 3);
```

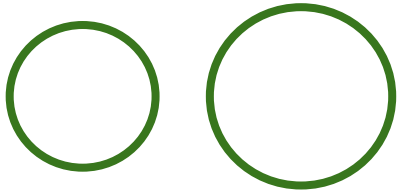
```
-bash-4.2$ ./lab06.out
lab06.out: lab06.cpp:60: void test(): Assertion `computeMinimum(2,3)==3' failed.
Aborted
-bash-4.2$
```

Today's Lab

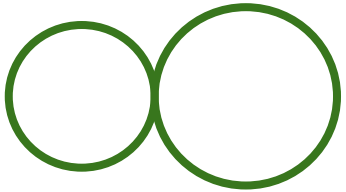
Exercise 1: basic selection/iteration warm-up

Exercise 2: get a user input, think of all test cases to meet, write test cases, write the function, check that you pass all test cases

Example test case 1: if 2 parks have centers far enough away relative to their sizes, they will have 0 points of intersection



Example test case 2: if the distance between 2 parks equals the sum of their radii, they have 1 point of intersection



Questions?