

1 C++: Lecture 3 on 31 May 2020

1.1 Loops

Loops are used to repeat blocks of code a certain number of times, or even forever. All loops utilize some sort of statements to determine when the set code should be run or not.

The syntax for a while loop is written as:

```
while(condition) {  
    code;  
}
```

For example, you can write code like so:

```
int n = 0;  
while (n <= 9) {  
    n = 10;  
}
```

Another way to stop a while loop is to use a break statement, a break statement will cause the while loop to determine.

```
n = 0;  
  
while(n <= 10){  
    //cout << "N is equal to: " << n << endl;  
    if(n == 7){  
        break;  
    }  
    n++;  
}
```

1.1.1 Task: Print All Numbers less than 100

Write some code that will print out all numbers that are less than 100. Take about 4 minutes to solve this.

Solution:

```
int num = 0  
while(num < 100) {  
    cout << num << endl;  
    num++;  
}
```

Similar to while loops, there also exists a do-while loops, these are like while loops but have a few differences (see lecture video). The syntax for a do-while loop is:

```
do {  
    code;  
} while(condition);
```

An example of this is:

```
int j = 98;  
do{  
    //cout << j << endl;  
    j = j - 2;  
} while(j > 50);
```

1.1.2 Task: Output all numbers from 100 to 1

```
int num = 100;  
  
do{  
    //cout << num << endl;  
    num--;  
} while(num > 1);
```

Another type of loop that exists is the for-loop, the syntax for a for loop is:

```
for(variable initializer, condition, action){  
    code;  
}
```

In a real example, this is seen as:

```
for(int j = 10; j > 0; j = j - 2){
    cout << j << endl;
}

cout << "End of the first for loop" << endl;

for(int i = 0; i < 20; i++){
    for(int j = 0; j < 10; j++){
        cout << i << endl;
    }
}
```

1.1.3 Task: Output even numbers

Take about 5 minutes to write a program that outputs all even numbers between 2 and 1000, use a for loop to accomplish this task.

Solution:

```
for(int i = 2; i <= 1000; i += 2){
    cout << i << endl;
}
```

Another type of loop is a forever loop, this is a loop that will never terminate.

```
while(1){
    cout << "Hi";
}

do{
    cout << "Hi";
} while(1);

for(;;) {
    cout << "Hi" << endl;
}
```

1.2 Arrays

Arrays are used to store data, you can define this by using the following syntax:

```
int a[number];
```

For example, you can define a numbers array with 5 slots by using:

```
int numbers[5];

// numbers[0] ... numbers[4] are all memory locations of the integer
// ↪ datatype, 5 total
```

All indices are zero-indexed, meaning the first index in an array is actually the zeroth index.

```
numbers[0] = 2342; // first element in numbers array takes value of 2,342
```

You can also have multidimensional arrays, such as 2 dimensional and 3 dimensional arrays, this is useful for complex data. For example, a vending machine is 3 dimensional because there are rows, columns, and depth.

```
int 2Dnums[5][5];
int 3Dnums[5][5][5];
```

You can make an array using any data type that you would like, for example:

```
char chars[123];
bool bools[89];
double doubles[908];
```

If you want to create an array and populate it with data, you can use loops to do this, for example:

```
int arr[1000];
int size = 1000;

for(int i = 0; i < 1000; i++){
    arr[i] = 45;
}

for(int i = 0; i < 1000; i++){
    cout << arr[i] << endl;;
}

int k = 0;
while(k < size){
    arr[k] = k;
    k++;
}

for(int i = 0; i < 1000; i++){
    cout << arr[i] << endl;;
}
```

1.2.1 Task: Array Creation

Take about 5 minutes to initialize and populate an array with a value like your favorite number.

Solution:

```
double fav[100];

for(int i = 0; i < 100; i++){
    fav[i] = 3.14;
}

for(int i = 0; i < 100; i++){
    //cout << fav[i] << endl;
}
```

1.3 Homework Review

1.3.1 Question 1

Write code that can serve as a automated cashier at a shop by asking customers what items they want and how many. This should be done using if/else statements, cin/cout, and strings. The code should output a list of items, have the user choose which ones and amount, and then the code should output more information about the item, i.e. the price.

```
cout << "Hello, Welcome to our store. What would you like to buy?" << endl
    ↪ ;

cout << "We have pizza(1), tacos(2), burgers(3), and fist(4). What would
    ↪ you like to buy?" << endl;

int x;
cin >> x;

int numToBuy;
double totalCost = 0;

if(x == 1){
    cout << "How many would you like to buy?";
    cin >> numToBuy;
    totalCost += numToBuy * 7.43;
}
if(x == 2){
    cout << "How many would you like to buy?";
    cin >> numToBuy;
    totalCost += numToBuy * 7.43;
}
if(x == 3){
    cout << "How many would you like to buy?";
    cin >> numToBuy;
    totalCost += numToBuy * 7.43;
}
if(x == 4){
    cout << "How many would you like to buy?";
    cin >> numToBuy;
    totalCost += numToBuy * 7.43;
}

cout << totalCost << " Thank You for shopping here" << endl;
```

1.3.2 Question 2:

Write code that can find the maximum of four inputed numbers, the minimum as well as the difference between the max/min.

```
int a, b, c, d;
cin >> a >> b >> c >> d;

int max = -100;
int min = 100;

if(a > max){
    max = a;
}
if(b > max){
    max = b;
}
if(c > max){
    max = c;
}
if(d > max){
    max = d;
}

if(a < min){
    min = a;
}
if(b < min){
    min = b;
}
}
```

```
if(c < min){  
    min = c;  
}  
if(d < min){  
    min = d;  
}  
  
cout << max << " " << min << endl;
```