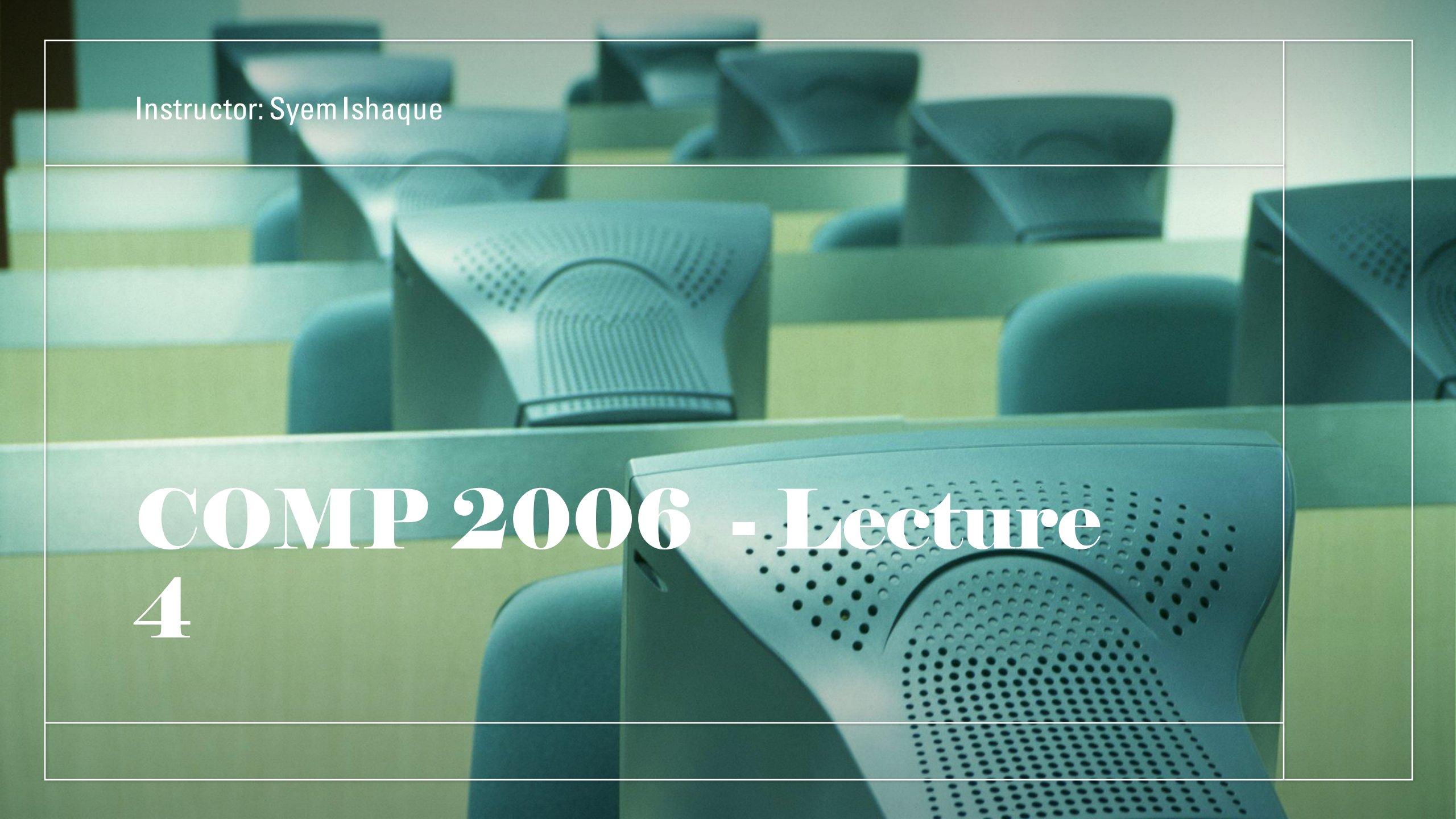

Instructor: Syem Ishaque

COMP 2006 - Lecture 4

Pointers

- Pointers are used to access objects in memory [1].
- Objects have a data type and address, pointers hold that address [1].
- Uses cases include dynamic memory allocation (change size of a data structure during runtime) and pass arguments by reference.
- Embedded programming requires the use of pointers, such as Fitbit.

Example of a pointer

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

int main() {

  char c = 's';
  char *p = &c;
  cout << "The value of pointer is: " << *p<< endl; // dereferencing a pointer
  cout << "The address is: " << &p << endl; // referencing a pointer
  return 0;

}
```

Returns
The value of pointer is: s The address is: 0x7ffee176f5f0

Changing values using a pointer

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

int main() {

  char c = 's';
  char *p = &c; // initializing pointer
  *p = 'I'; // changing the value of c using a pointer

  cout << "The value of c is: " << c << endl;
  return 0;
}
```

Returns
The value of c is: I

Referencing

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

int main() {

  int x = 18;
  int &y = x; // referencing

  y = 21; // changing value
  cout << "The value of x is: " << x << endl;
  return 0;
}
```

Returns
The value of x is: 21

Constant Reference

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

int main() {

  int x = 18;
  const int &y = x; // constant with x, y can't change values, only x can!
  x = 21; // changing value
  cout << "The value of y is: " << y << endl;
  return 0;

}
```

Returns
The value of y is: 21

Conditional Statements

- Executed to meet conditions.
- If a certain condition is true, then the program will execute a certain statement, otherwise it will execute another statement if its false.
- If something occurs, the program will execute a certain function.

If and Else statements

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

int main() {

  int x = 17;
  if (x == 18){ // executed if x is 18
    cout << "The value of x is: " << x << endl;
  }
  else { // executed if x is not 18
    cout << " The value of x is not 18, its: " << x << endl;
  }
  return 0;
}
```

Returns
The value of x is not 18, its: 17

If and Else using boolean

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

int main() {

bool a = false;
  if (a){ // executed if true
    cout << "The statement is true " << endl;
  }
  else { // executed if false
    cout << " The statement is false: " << endl;
  }
  return 0;
}
```

Returns
The statement is false

Switch Statement

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

int main() {
  int x = 1;
  switch (x) // checks value
  {
  case 0:
    cout << "My fav type of music is hip-hop" << endl;
    break;
  case 1: // executes this since it meets value
    cout << "My fav type of music is melodious" << endl;
    break;
  default:
    cout << "I don't like music" << endl;
    break;
  }
```

Returns
My fav type of music is melodious

Takeways

- Learned about pointers and conditional statements.
- How can we use this for real world applications?

Reference

- 1. Modern C++ for Absolute Beginners: A Friendly Introduction to C++ Programming Language and C++11 to C++20 Standards