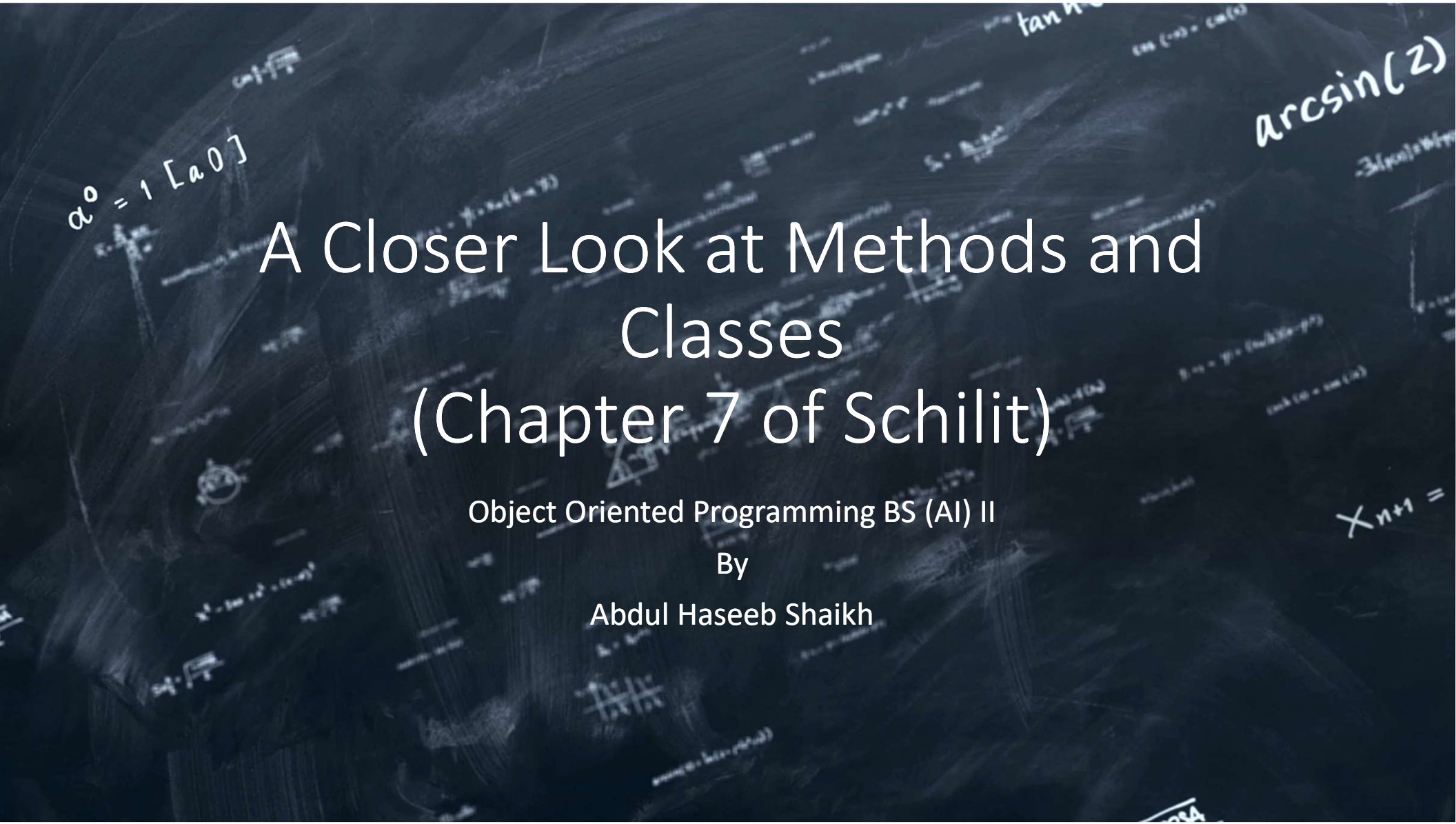

A Closer Look at Methods and Classes (Chapter 7 of Schilit)

Object Oriented Programming BS (AI) II

By

Abdul Haseeb Shaikh

Revision of Methods/Functions

- What is a method/function?
- Parameter vs Argument
- Void?
  - Why do we use these methods

Polymorphism

OOP Principle

When an object of class shows multiple behaviors.

Overloading Methods

Two or more methods in same class:

Having same name

But different return type

Different type/number of arguments

Type or number of argument determine the method to be called

Overloading Methods

- Return type alone is not sufficient during the call of method
  - Java matches the arguments with the parameters to call a particular method

Overloading Methods

```
// Demonstrate method overloading.
class OverloadDemo {
  void test() {
    System.out.println("No parameters");
  }

  // Overload test for one integer parameter.
  void test(int a) {
    System.out.println("a: " + a);
  }

  // Overload test for two integer parameters.
  void test(int a, int b) {
    System.out.println("a and b: " + a + " " + b);
  }

  // Overload test for a double parameter
  double test(double a) {
    System.out.println("double a: " + a);
    return a*a;
  }
}
```

What should be the output now???

```
class Overload {
  public static void main(String args[]) {
    OverloadDemo ob = new OverloadDemo();
    double result;

    // call all versions of test()
    ob.test();
    ob.test(10);
    ob.test(10, 20);
    result = ob.test(123.25);
    System.out.println("Result of ob.test(123.25): " + result);
  }
}
```

Example of Automatic Type conversion

```
// Automatic type conversions apply to overloading.
class OverloadDemo {
  void test() {
    System.out.println("No parameters");
  }

  // Overload test for two integer parameters.
  void test(int a, int b) {
    System.out.println("a and b: " + a + " " + b);
  }

  // Overload test for a double parameter
  void test(double a) {
    System.out.println("Inside test(double) a: " + a);
  }
}
```

Example of Automatic Type conversion

```
class Overload {
  public static void main(String args[]) {
    OverloadDemo ob = new OverloadDemo();
    int i = 88;

    ob.test();
    ob.test(10, 20);

    ob.test(i); // this will invoke test(double)
    ob.test(123.2); // this will invoke test(double)
  }
}
```

Overloading Methods

Advantage is common name methods for different activities of same nature, exp: Addition

Those languages which don’t support overloading(like C):

Give unique names for methods

abs() for int and labs() for long int

It is one of the ways in which java can achieve polymorphism:

One object shows multiple behaviors

Overloading Constructors

- Requires three parameters

```
class Box {
  double width;
  double height;
  double depth;

  // This is the constructor for Box.
  Box(double w, double h, double d) {
    width = w;
    height = h;
    depth = d;
  }

  // compute and return volume
  double volume() {
    return width * height * depth;
  }
}
```

Overloading Constructors

- This is invalid right now

```
Box ob = new Box();
```

- What if you wanted a box without parameters