

---

## LAB 02 HANDOUT AND TASKS

---

Prepared by:

Md. Jubair Ibna Mostafa

Assistant Professor, IUT CSE



Department of Computer Science and Engineering  
Islamic University of Technology

## Contents

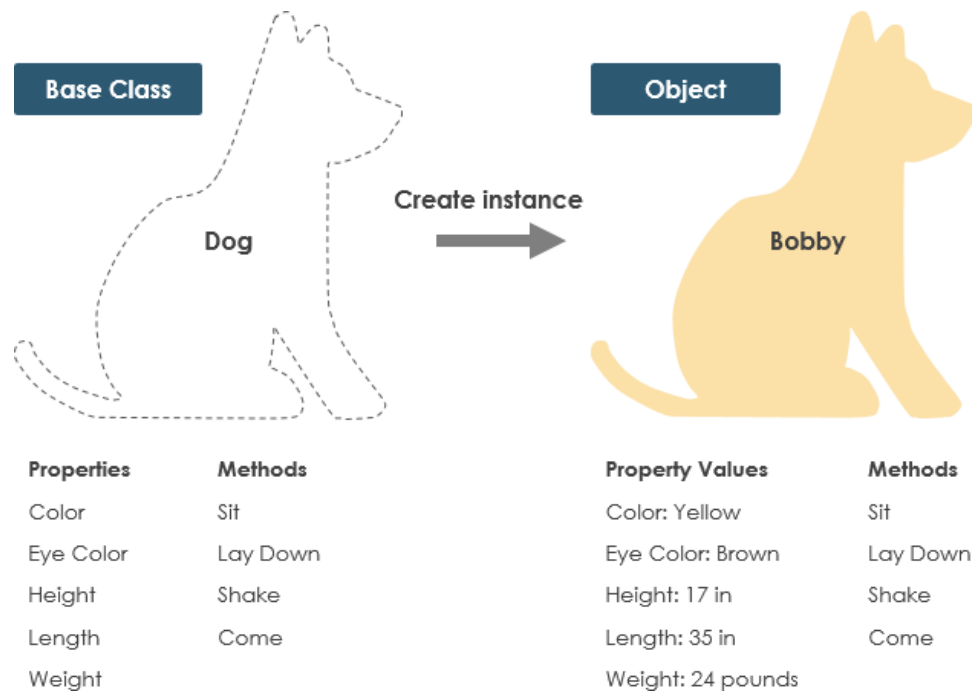
|          |                           |                   |
|----------|---------------------------|-------------------|
| <b>1</b> | <b>Class &amp; Object</b> | <b>3</b>          |
| 1.1      | Code Example . . . . .    | 3                 |
| <b>2</b> | <b>Task</b>               | <b>Marks 10 6</b> |

# 1 CLASS & OBJECT

Class is a template for an object and an object is an instance of a class.

Class is just a logical framework, the object is the physical reality.

**Figure 1: Class and Object**



## 1.1 Code Example

```

1 class Dog {
2     String color;
3     String eyeColor;
4     double height;
5     double length;
6     double weight;
7
8     Dog(String color, String eyeColor, double height, double length,
9         double weight) {
10         this.color = color;
11         this.eyeColor = eyeColor;

```

```
11     this.height = height;
12     this.length = length;
13     this.weight = weight;
14 }
15
16 void sit() {
17     System.out.println("The dog is sitting.");
18 }
19
20 void layDown() {
21     System.out.println("The dog is laying down.");
22 }
23
24 void shake() {
25     System.out.println("The dog is shaking hands.");
26 }
27
28 void come() {
29     System.out.println("The dog is coming.");
30 }
31
32 // Method to check if the dog is overweight
33 void checkWeight() {
34     if (weight > 20.0) {
35         System.out.println("The dog is overweight.");
36     } else {
37         System.out.println("The dog is at a healthy weight.");
38     }
39 }
40
41 // Method to check if the dog is tall
42 void checkHeight() {
43     if (height > 0.6) {
44         System.out.println("The dog is tall.");
45     } else {
```

```
46         System.out.println("The dog is of average height.");
47     }
48 }
49
50 // Main method for testing
51 public static void main(String[] args) {
52     Dog bobby = new Dog("yellow", "brown", 17.0, 35.0, 24.0);
53     bobby.sit();
54     bobby.layDown();
55     bobby.shake();
56     bobby.come();
57     bobby.checkWeight(); // Check if the dog is overweight
58     bobby.checkHeight(); // Check if the dog is tall
59 }
60 }
```

A constructor is a special method that is used to initialize objects of a class. It has the same name as the class and does not have a return type, not even void. Constructors are called automatically when an object of a class is created, and they can be used to perform initialization tasks such as setting initial values for object properties.

The objective of having multiple constructors in a class is to provide flexibility in object creation by allowing different ways to initialize objects. Each constructor can take different parameters or a different number of parameters, allowing the caller to choose the most appropriate constructor based on the context or available information.

Having multiple constructors can be useful for the following reasons:

1. **Parameter flexibility:** Different constructors can accept different combinations of parameters, providing flexibility in how objects are initialized.
2. **Default values:** Constructors can provide default values for properties if certain parameters are not provided, making object creation more convenient.
3. **Initialization logic:** Constructors can contain initialization logic that is specific to different scenarios, allowing for more complex object creation processes.

```
1 // Constructor with color and eyeColor
2   Dog(String color, String eyeColor) {
3       this.color = color;
4       this.eyeColor = eyeColor;
5       this.heightInInches = 0.0;
6       this.lengthInInches = 0.0;
7       this.weightInPounds = 0.0;
8   }
9
10  // Constructor with height, length, and weight
11  Dog(double heightInInches, double lengthInInches, double
12  weightInPounds) {
13      this.color = "";
14      this.eyeColor = "";
15      this.heightInInches = heightInInches;
16      this.lengthInInches = lengthInInches;
17      this.weightInPounds = weightInPounds;
18  }
```

## 2 TASK

## MARKS 10

You are required to design and implement a system for managing information about students, courses, and grades using object-oriented programming concepts. The system should consist of three classes: Student, Course, and Grade.

The **Student** class should store details such as the student's name, age, and ID. It should have multiple constructors to accommodate different ways of initializing student objects, including constructors with and without parameters.

The **Course** class should represent individual courses and store attributes such as course code and name. It should also have multiple constructors to accommodate different ways of initializing course objects, including constructors with and without parameters.

The **Grade** class should handle grading information, including attendance marks (out of 30), quiz marks (out of 45), mid-term marks (out of 75), and final exam marks (out of 150).

It should have multiple constructors to accommodate different ways of initializing grade objects, including constructors with and without parameters.

Each class should have a `display` method that prints the relevant information. Additionally, the **Grade** class should have a `displayGrades` method that calculates the percentage and letter grade based on the given grading system and displays this information.

The grading system is as follows:

- A+: 80% and above
- A: 75% to below 80%
- A-: 70% to below 75%
- B+: 65% to below 70%
- B: 60% to below 65%
- B-: 55% to below 60%
- C+: 50% to below 55%
- C: 45% to below 50%
- D: 40% to below 45%
- F: below 40%

In the **Main** class, you are required to create instances of the **Student**, **Course**, and **Grade** classes, invoke their `display` methods to print the details, and then invoke the `displayGrades` method of the **Grade** class to display the calculated grades.