



# OBJECT ORIENTED PROGRAMMING

Abdullah Bin Kasem Bhuiyan  
Topic 2: Data types, declarations,  
and variables in Java

# SECTIONS OF A JAVA PROGRAM

- ❖ Documentation Section
- ❖ Package Declaration
- ❖ Import Statements
- ❖ Interface Section
- ❖ Class Definition
- ❖ Class Variables and Variables
- ❖ Main Method Class
- ❖ Methods and Behaviors

# DOCUMENTATION SECTION

The documentation section is an important section but optional for a Java program. It includes **basic information** about a Java program. The information includes the **author's name, date of creation, version, program name, company name,** and **description** of the program.

Example:

- **Single-line Comment:** It starts with a pair of forwarding slash (`//`). For example:

1. `//First Java Program`

- **Multi-line Comment:** It starts with a `/*` and ends with `*/`. We write between these two symbols. For example:

1. `/*It is an example of`

2. `multiline comment*/`

- **Documentation Comment:** It starts with the delimiter (`/**`) and ends with `*/`. For example:

1. `/**It is an example of documentation comment*/`

# PACKAGE DECLARATION

The package declaration is optional. It is placed just after the documentation section. In this section, we declare the **package name** in which the class is placed. Note that there can be **only one package** statement in a Java program. It must be defined before any class and interface declaration. It is necessary because a Java class can be placed in different packages and directories based on the module they are used.

Example:

**1.package** cisbatch7; //where cisbatch7 is the package name

**2.package** bd.edu.diu.cis.batch7; //where .bd is the root directory and batch7 is the subdirectory

# IMPORT STATEMENTS

The package contains the many predefined classes and interfaces. If we want to use any class of a particular package, we need to import that class. The import statement represents the class stored in the other package. We use the **import** keyword to import the class. It is written before the class declaration and after the package statement. We use the import statement in two ways, either import a specific class or import all classes of a particular package.

Example:

**1.import** java.util.Scanner; //it imports the Scanner class only

**2.import** java.util.\*; //it imports all the class of the java.util package

# INTERFACE SECTION

It is an optional section. We can create an **interface** in this section if required. We use the **interface** keyword to create an interface.

An interface is a slightly different from the class. It contains only **constants** and **method** declarations. Another difference is that it cannot be instantiated.

Example:

```
1.interface car
2.{
3.void start();
4.void stop();
5.}
```

# CLASS DEFINITION

In this section, we define the class. It is **vital** part of a Java program. Without the class, we cannot create any Java program. A Java program may contain more than one class definition. We use the **class** keyword to define the class. The class is a blueprint of a Java program.

Example:

```
1.class Student //class definition
```

```
2.{
```

```
3.}
```

# CLASS VARIABLES AND CONSTANTS

In this section, we define variables and **constants** that are to be used later in the program. In a Java program, the variables and constants are defined just after the class definition. The variables and constants store values of the parameters. It is used during the execution of the program.

Example:

```
1.class Student //class definition
2.{
3.String sname; //variable
4.int id;
5.double percentage;
6.}
```

# MAIN METHOD CLASS

In this section, we define the **main() method**. It is essential for all Java programs. Because the execution of all Java programs starts from the main() method. In other words, it is an entry point of the class. It must be inside the class. Inside the main method, we create objects and call the methods.

Example:

```
1.public static void main(String args[])  
2.{  
3.}
```

# MAIN METHOD EXAMPLE

```
1.public class Student //class definition
2.{
3.public static void main(String args[])
4.{
5.//statements
6.}
7.}
```

# METHODS AND BEHAVIOR

In this section, we define the functionality of the program by using the methods. The methods are the set of instructions that we want to perform. These instructions execute at runtime and perform the specified task.

Example:

```
public class Demo //class definition
{
    public static void main(String args[])
    {
        void display()
        {
            System.out.println("Welcome to javatpoint");
        }
    }
    //statements
}
}
```

# IDE WE WILL USE

1. any text editor
2. eclipse
3. netbeans