

ID:11757

NAME:SALMAN KHAN

SUBJECT:OBJECT ORIENTED PROGRAMMING

TEACHER:M.AYUB KHAN

DATE:26.08.2020

**ANS1:CLASS:** a class describes the contents of the objects that belong to it: it describes an aggregate of data fields (called instance variables), and defines the operations (called methods).a class describes the contents of the objects that belong to it: it describes an aggregate of data fields (called instance variables), and defines the operations (called methods).A class can be defined as a template/blueprint that describes the behavior/state that the object of its type support.A class is a user defined blueprint or prototype from which objects are created. It represents the set of properties or methods that are common to all objects of one type. In general, class declarations can include these components, in order:

**Modifiers :** A class can be public or has default access (Refer this for details).

**Class name:** The name should begin with a initial letter (capitalized by convention).

**Superclass(if any):** The name of the class's parent (superclass), if any, preceded by the keyword extends. A class can only extend (subclass) one parent.

**Interfaces(if any):** A comma-separated list of interfaces implemented by the class, if any, preceded by the keyword implements. A class can implement more than one interface.

**Body:** The class body surrounded by braces, { }.

Constructors are used for initializing new objects. Fields are variables that provides the state of the class and its objects, and methods are used to implement the behavior of the class and its objects.

There are various types of classes that are used in real time applications such as nested classes, anonymous classes, lambda expressions.

**EXAMPLE: simple class:**

```

public class Dog {
    String breed;
    int age;
    String color;

    void barking() {
    }

    void hungry() {
    }

    void sleeping() {
    }
}

```

A class can contain any of the following variable types.

- **Local variables** – Variables defined inside methods, constructors or blocks are called local variables. The variable will be declared and initialized within the method and the variable will be destroyed when the method has completed.
- **Instance variables** – Instance variables are variables within a class but outside any method. These variables are initialized when the class is instantiated. Instance variables can be accessed from inside any method, constructor or blocks of that particular class.
- **Class variables** – Class variables are variables declared within a class, outside any method, with the static keyword.

A class can have any number of methods to access the value of various kinds of methods. In the above example, barking(), hungry() and sleeping() are methods.

Following are some of the important topics that need to be discussed when looking into classes of the Java Language.

**OBJECT:** an object is an element (or instance) of a class; objects have the behaviors of their class. The object is the actual component of programs, while the class specifies how instances are created and how they behave.●

Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behaviors – wagging the tail, barking, eating. An object is an instance of a class. It is a basic unit of Object Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods. An object consists of :

**State :** It is represented by attributes of an object. It also reflects the properties of an object.

**Behavior** : It is represented by methods of an object. It also reflects the response of an object with other objects.

**Identity** : It gives a unique name to an object and enables one object to interact with other objects.

As mentioned previously, a class provides the blueprints for objects. So basically, an object is created from a class. In Java, the new keyword is used to create new objects.

There are three steps when creating an object from a class –

- **Declaration** – A variable declaration with a variable name with an object type.
- **Instantiation** – The 'new' keyword is used to create the object.
- **Initialization** – The 'new' keyword is followed by a call to a constructor. This call initializes the new object.

Following is an example of creating an object –

## EXAMPLE:

```
public class Puppy {
    public Puppy(String name) {
        // This constructor has one parameter, name.
        System.out.println("Passed Name is :" + name );
    }

    public static void main(String []args) {
        // Following statement would create an object myPuppy
        Puppy myPuppy = new Puppy( "tommy" );
    }
}
```

If we compile and run the above program, then it will produce the following result

## Output

```
Passed Name is :tommy
```

**ANS2:** # Python Program to print table

# of a number given range

```
def table(n, r):
```

```
    for i in range (1, r + 1):
```

```
        # multiples from 1 to r (range)
```

```
        print "%d * %d = %d" % (n, i, n * i)
```

```
# number for which table is evaluated
```

```
n = 8
```

```
# range upto which multiples are to be calculated
```

```
r = 12
```

```
table(n,r)
```

## OUTPUT:

```
8 * 1 = 8
```

```
8 * 2 = 16
```

```
8 * 3 = 24
```

```
8 * 4 = 32
```

```
8 * 5 = 40
```

```
8 * 6 = 48
```

```
8 * 7 = 56
```

```
8 * 8 = 64
```

```
8 * 9 = 72
```

```
8 * 10 = 80
```

```
8 * 11 = 88
```

```
8 * 12 = 96
```

## ANS3:

```
# create a Car object
```

```
    car = Car("1990 Laser", "Plymouth")
```

```
    # calls the accelerate method
```

```
    print "Accelerating . . ."
```

```
    for i in xrange(5):
```

```
        car.accelerate()
```

```
        print car.get_speed()
```

```
# calls the brake method five times.  
print "Decelerating . . ."  
for i in xrange(5):  
    car.brake()  
    print car.get_speed()  
if __name__ == "__main__":  
    main()
```