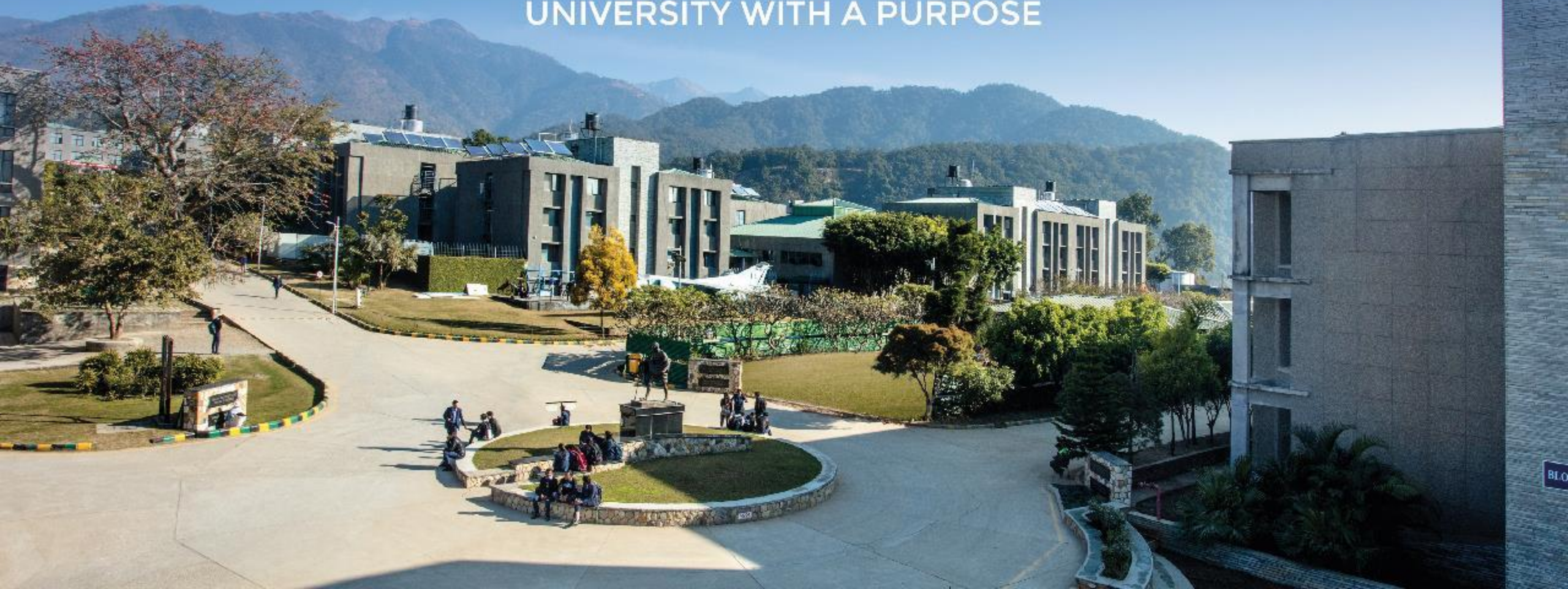
UPES

UNIVERSITY WITH A PURPOSE



UPES
UNIVERSITY WITH A PURPOSE

Object Oriented Programming

Anonymous Inner classes in Java

- Anonymous classes enable you to make your code more concise. They enable you to declare and instantiate a class at the same time.
- They are like local classes except that they do not have a name. Use them if you need to use a local class only once.
- While local classes are class declarations, anonymous classes are expressions, which means that you define the class in another expression.
- Anonymous inner classes are declared without any name at all. They are created in two ways.

Anonymous Inner classes in Java

a) As subclass of specified type

```
class Demo {
void show() {
        System.out.println(“Hello");
}
}
class InnerDemo {

// An anonymous class with Demo as base class
static Demo d = new Demo() {
        void show() {
                super.show();
                System.out.println(“Hi");
        }
};
public static void main(String[] args){
        d.show();
}
}
```

Output:

```
Hello
Hi
```

Anonymous Inner classes in Java

b) As implementer of the specified interface

```
interface Hello {
        void show();
}
class InnerDemo {
        // An anonymous class that implements Hello interface
        static Hello h = new Hello() {
                public void show() {
                        System.out.println("anonymous inner class");
                }
        };
        public static void main(String[] args) {
                h.show();
        }
}
```

Output:

```
anonymous inner class
```

Note: In above code we create an object of anonymous inner class but this anonymous inner class is an implementer of the interface Hello. Any anonymous inner class can implement only one interface at one time. It can either extend a class or implement interface at a time.

Syntax of Anonymous Classes

- As mentioned previously, an anonymous class is an expression. The syntax of an anonymous class expression is like the invocation of a constructor, except that there is a class definition contained in a block of code.

- Consider the instantiation of the `frenchGreeting` object in the example.

```
HelloWorld frenchGreeting = new HelloWorld()
{
            String name = “hi”;
            public void greet()
            {
                greetSomeone(“hi”);
            }
            public void greetSomeone(String someone)
            {
                name = someone;
                System.out.println(“hello” + name);
            }
};
```

Syntax of Anonymous Classes

The anonymous class expression consists of the following:

- The new operator
- The name of an interface to implement or a class to extend. In this example, the anonymous class is implementing the interface Hello.
- Parentheses that contain the arguments to a constructor, just like a normal class instance creation expression.

Note: When you implement an interface, there is no constructor, so you use an empty pair of parentheses, as in this example.

- A body, which is a class declaration body.

Because an anonymous class definition is an expression, it must be part of a statement. In this example, the anonymous class expression is part of the statement that instantiates the frenchGreeting object. (This explains why there is a semicolon after the closing brace.)

References

1. Accessed from https://www.geeksforgeeks.org/local-inner-class-java/
2. Accessed from https://docs.oracle.com/javase/tutorial/java/javaOO/nested.html

THANK YOU

UPES

UNIVERSITY WITH A PURPOSE