HNDIT3012 - OBJECT ORIENTED PROGRAMMING

LECTURE 08 – ABSTRACT CLASS

ABSTRACTION

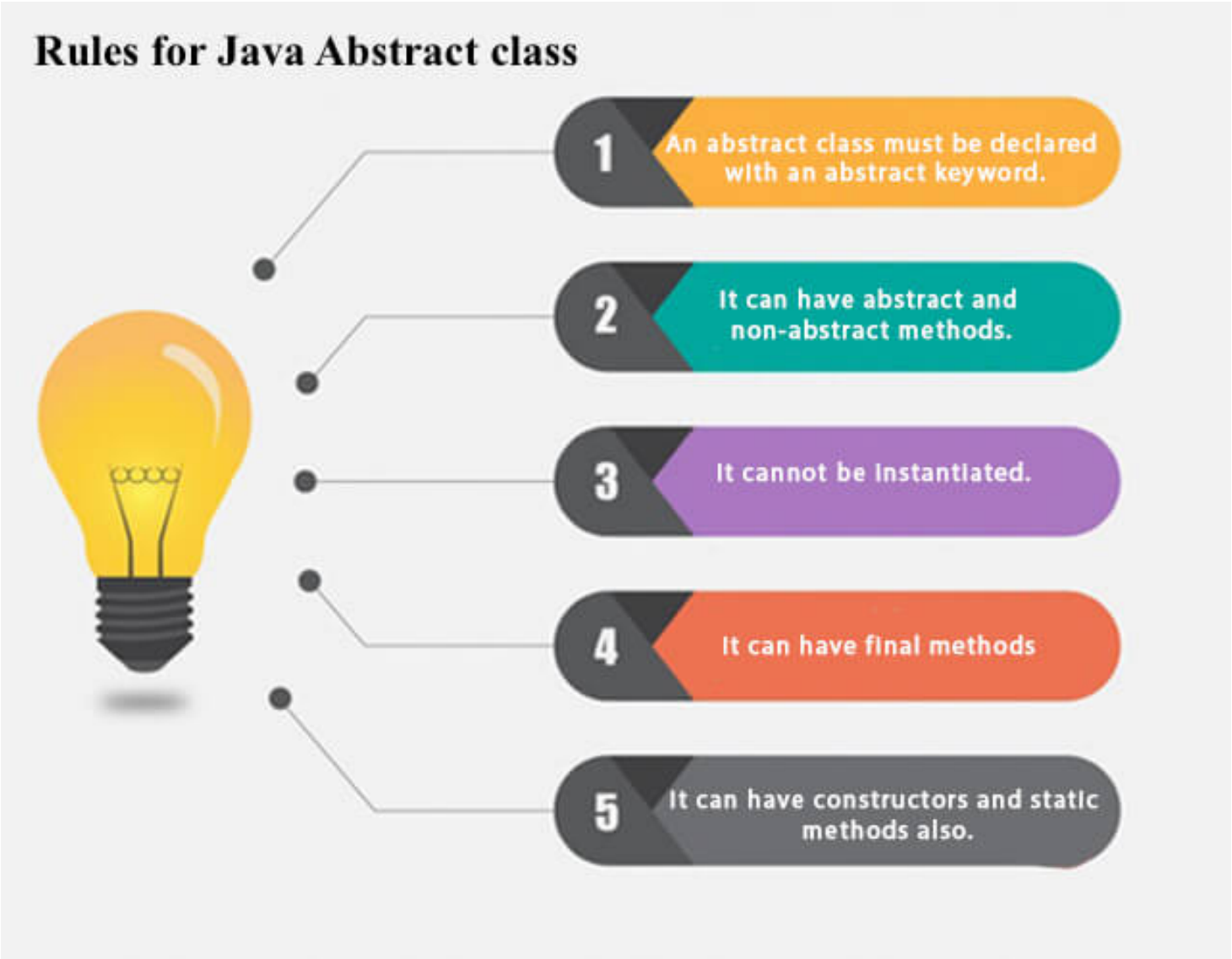
- **Abstraction** is the process of hiding certain details and only show the essential features of the object.
- Hiding the implementation details from the user
- Filter out /hiding unnecessary information
- Advantages
  - Hiding implementation complexity from user
  - Object is easy to used by users

ABSTRACT CLASS

- An abstract class is a class that **cannot be instantiated** and can contain both abstract and non-abstract methods.
- It needs to be extended (sub classed)
- In an abstract class, some methods can be implemented, while others are left abstract, meaning that they have no implementation and must be **overridden** by concrete **subclasses**.
- Abstract classes can have access modifiers such as public, protected, and private for their methods and properties
- Abstract class can have final, non-final, static and non-static variables

FEATURES OF ABSTRACT CLASS

- Cannot be instantiated
- Contains at least one pure virtual function
- Can contain both abstract and non-abstract methods
- Can have constructors and destructors
- Can have member variables
- Can be used as a base class


Rules for Java Abstract class
1
An abstract class must be declared with an abstract keyword.
2
It can have abstract and non-abstract methods.
3
It cannot be instantiated.
4
It can have final methods
5
It can have constructors and static methods also.

EXAMPLE

```
abstract class person {

    abstract void printInfo(); // no body need to implement in a subclass
}

class student extends person {
    void printInfo() // @ Override abstract class
    {
        String name = “Janaka";
        int age = 21;
        float gpa = 3.47;

        System.out.println(name);
        System.out.println(age);
        System.out.println(gpa);
    }
}
```

```
class base {
    public static void main(String args[])
    {
        person s = new student();
        s.printInfo(); // call override method
    }
}
```

EXAMPLE 02

```
abstract class Vehicle
	{
	abstract void run();
	}
class Bike extends Vehicle
	{
	void run()
		{
		System.out.println("Bike running safely..");
		}
	public static void main(String args[])
		{
		Vehicle obj = new Bike ();
		obj.run();
		}
	}
```

An abstract class can have data member, abstract method, method body, constructor and even main() method

EXAMPLE 03

```
abstract class Bike{
    abstract void run();
}

class Honda4 extends Bike{
        void run(){
            System.out.println("running safely");
        }

    public static void main(String args[]){
        Bike obj = new Honda4();
        obj.run();
    }
}
```