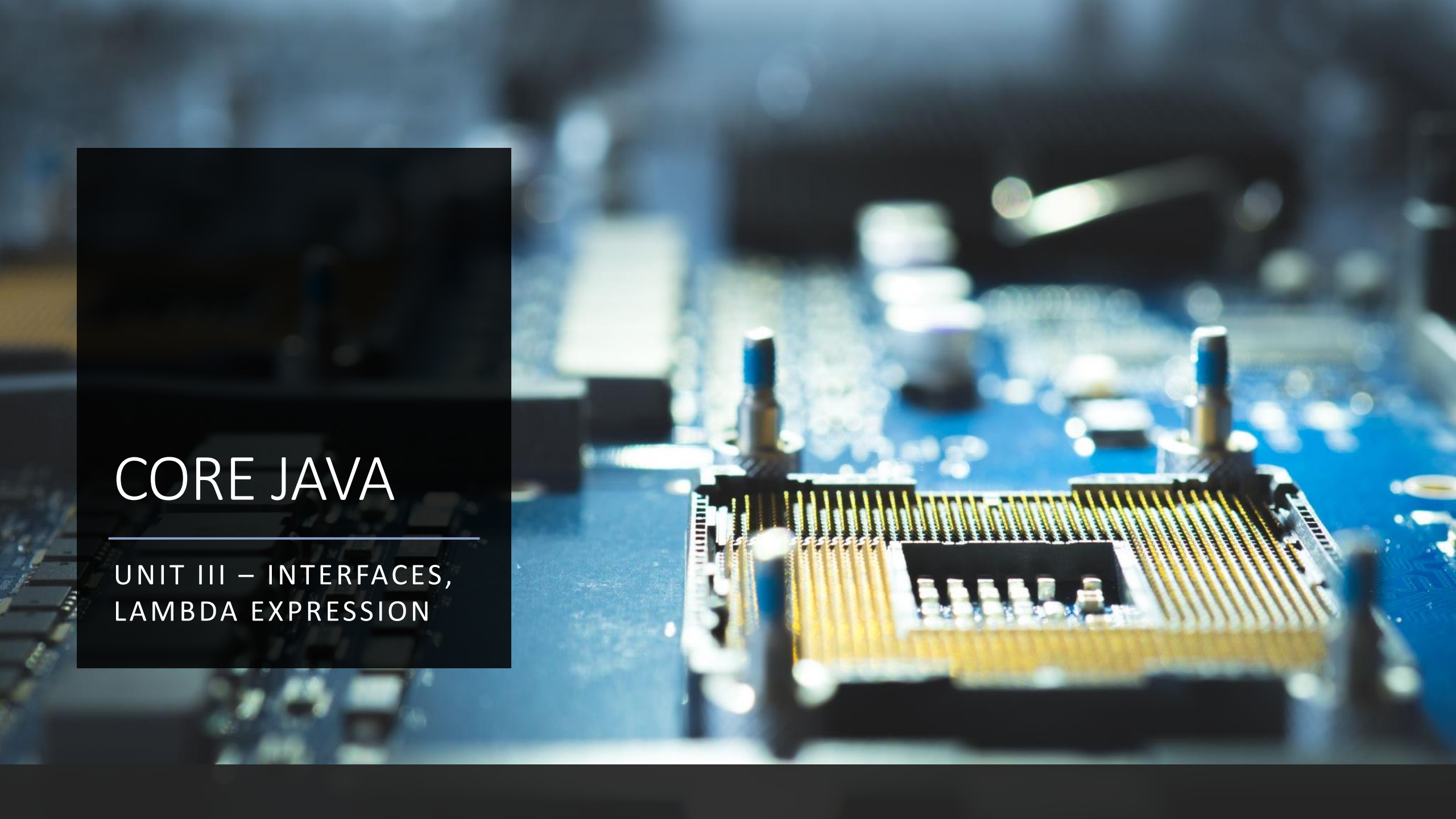

CORE JAVA

UNIT III – INTERFACES, LAMBDA EXPRESSION

Match Card Activity

MATCH: CORE JAVA | QUIZLET

Identify error in below given code snippets

1. Interface Student{….}

   class Result extends Student{...}

2. class Employee{

   abstract void getEmp(); }

3 . class B{

   final void getDim(){…..}

   }

4. class Manager{

   final void getData(){…..}

   }

   class Executive extends Manager{

   void getData(){….}

   }

5. interface myinterface{

   int x; }

NESTED INTERFACES

An interface can be declared as a member of the class or another interface.

Called as member interface or nested interface.

Can be declared as public, private or protected.

EXTENDING INTERFACES

Keyword *extends* is used.

When a class implements an interface that inherits another interface, it must provide implementations for all methods defined within the inheritance chain.

An interface can extend multiple interfaces too.

Ex:

```
interface A{
        void m1();
        void m2()
}
interface B extends A{
        void m3();
}
```

LAMBDA EXPRESSION

Provides a clear and concise way to represent one method interface(functional interface) using an expression.

Saves a lot of code as we just write the implementation code here.

Syntax:

```
(argument-list) -> {body}
```

EXAMPLE:

```
(int x) -> {System.out.println(2*x);}
```

TYPES OF LAMBDA EXPRESSION

Zero Parameter

- () -> {System.out.println("Zero Parameter lambda");}

One Parameter

- (p) -> {System.out.println("One Parameter:"+p);}

Multiple Parameter

- (p1,p2) -> {System.out.println("Multiple parameters: "+p1+","+p2);}


Thank You!