

Group A

Lab 3B

Scala

Participants:

1. Aadarsha chapagain
- 2.Jyoti shukla
- 3.Rishi Phaneendra Varma
- 4.Priti Bhale
- 5.Sreya Treesa Johnny
- 6.Piyush Bhatia

Slide 3:

“ public int age();” is a getter (accessor method)

“public void age_=eq(int);” is a setter . _\$eq means _equal to
public Person(\$iw); : constructor of the class

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Left Sidebar:** Package Explorer (BDM_1024_Scala_asgn_part1).
- Center Area:** Problems, Tasks, Console (BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)).

```
scala> class Person
           |     var age = 20
           |
defined class Person

scala> :javap -private Person
Compiled from "Person.java"
public class $iwcm$BDM_1024_Scala_asgn_part1$Person {
    private int age;
    public int age();
    public void age_=eq(int);
    public $iwcm$Person();
}

scala> val person = new Person()
person: Person@18998560

scala> val person = new Person
person: Person@5fdd97c1
```
- Right Sidebar:** Outline (An outline is not available), Group A (Participants list).
- Bottom Status Bar:** 693M of 933M, 11°C Mostly cl..., 1:14 PM, 1:14 PM.

Slide 4:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Left Sidebar:** Package Explorer (BDM_1024_Scala_asgn_part1).
- Center Area:** Problems, Tasks, Console (BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:26 PM)).

```
scala> person: Person = Person@10998560

scala> val person = new Person
person: Person@5fdd97c1

scala> person.age
res0: Int = 20

scala> person.age_=eq(21)
scala> person.age
res1: Int = 21

scala> person.age_=eq(53)
scala> person.age
res4: Int = 53

scala>
```
- Right Sidebar:** Outline (An outline is not available), Group A (Participants list).
- Bottom Status Bar:** 299M of 992M, 11°C Mostly cl..., 1:20 PM, 1:20 PM.

Slide 5:

Here the val is used so it won't allow to set the variable so error is thrown

The screenshot shows a Windows desktop with three windows open. On the left is the 'workspace - Scala IDE' window, which contains a Scala interpreter console. The code defines a class Person with a private final val age. A val person is then created, and its age is set to 20. This results in an error message: 'error: value age_ is not a member of Person'. On the right is a 'BDM_Group...' window titled 'Group A' showing a list of participants: 1. Aadarsha chapagain, 2. Jyoti shukla, 3. Rishi Phaneendra Varma, 4. Priti Bhole, 5. Sreya Treesa Johny, and 6. Piyush Bhatia. The system tray at the bottom indicates it's 11°C, mostly cloudy, and the time is 1:24 PM.

```
scala> class Person {  
    | val age = 20  
    | }  
defined class Person  
  
scala> :javap -private Person  
Compiled from "console"  
public class $line11.$read$$iw$Person {  
    private final int age;  
    public int age();  
    public $line11.$read$$iw$Person();  
}  
  
scala> val person = new Person  
person: Person = Person@444428583  
  
scala> person.age  
res5: Int = 20  
  
scala> person.age_ = 40  
  
<console>:12: error: value age_ is not a member of Person  
      person.age_ = 40  
  
<console>:13: error: value age_ is not a member of Person  
val $ires0 = person.age_  
          ^  
  
scala>
```

Slide 6:

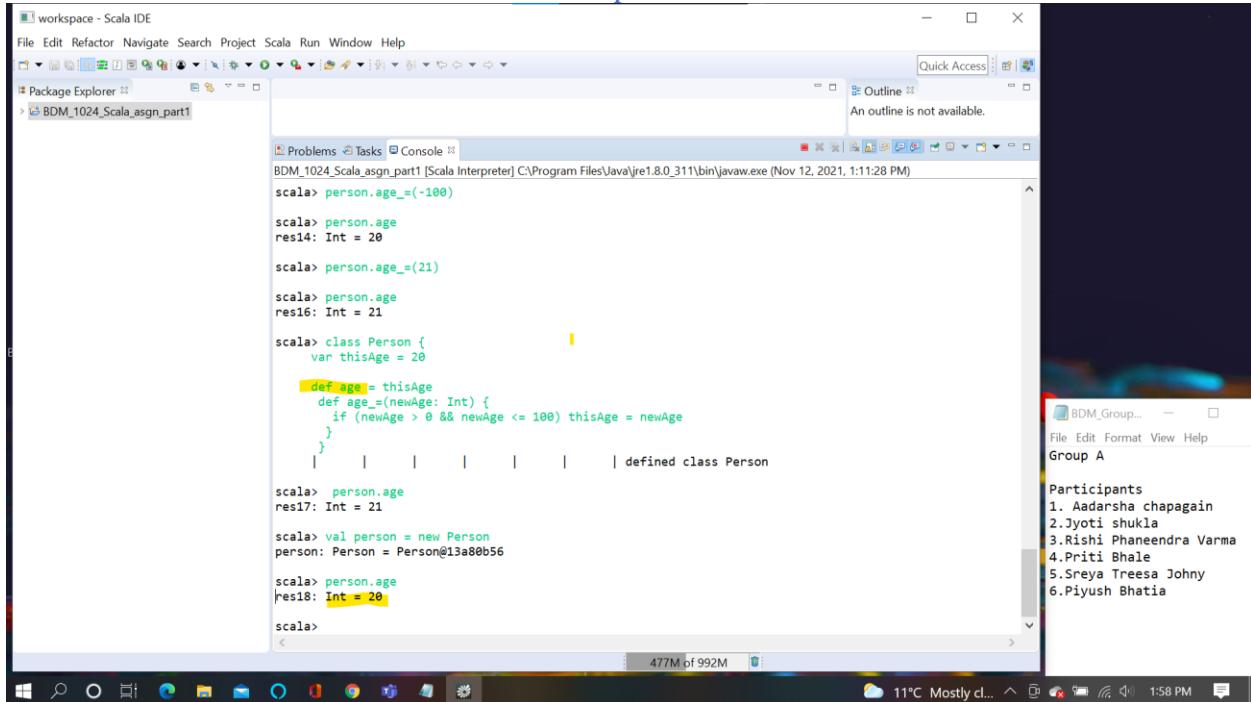
We have written our own setter method with condition to set if the age lies between 0 to 100

The screenshot shows a Windows desktop with three windows open. On the left is the 'workspace - Scala IDE' window, which contains a Scala interpreter console. The code defines a class Person with a private var age. It includes a getter def age() = this.age and a setter def age_(newAge: Int) { if (newAge > 0 && newAge <= 100) thisAge = newAge }. A val person is created, and its age is set to 20. Then, an attempt to set the age to 100 fails because the condition in the setter is not met. On the right is a 'BDM_Group...' window titled 'Group A' showing a list of participants: 1. Aadarsha chapagain, 2. Jyoti shukla, 3. Rishi Phaneendra Varma, 4. Priti Bhole, 5. Sreya Treesa Johny, and 6. Piyush Bhatia. The system tray at the bottom indicates it's 11°C, mostly cloudy, and the time is 1:51 PM.

```
scala> person.age_=-100  
scala> class Person {  
    var thisAge = 0  
    def age() = thisAge  
    def age_(newAge: Int) {  
        if (newAge > 0 && newAge <= 100) thisAge = newAge  
    }  
}  
  
scala>  
  
scala> val person = new Person  
person: Person = Person@95aa646e  
  
scala> person.age_  
res12: Int = 20  
  
scala> person.age_=-100  
scala> person.age_  
res14: Int = 20  
  
scala> person.age_=(21)  
scala> person.age_  
res16: Int = 21  
  
scala>
```

Slide 7:

Scala allows to use setter method even without parentheses.



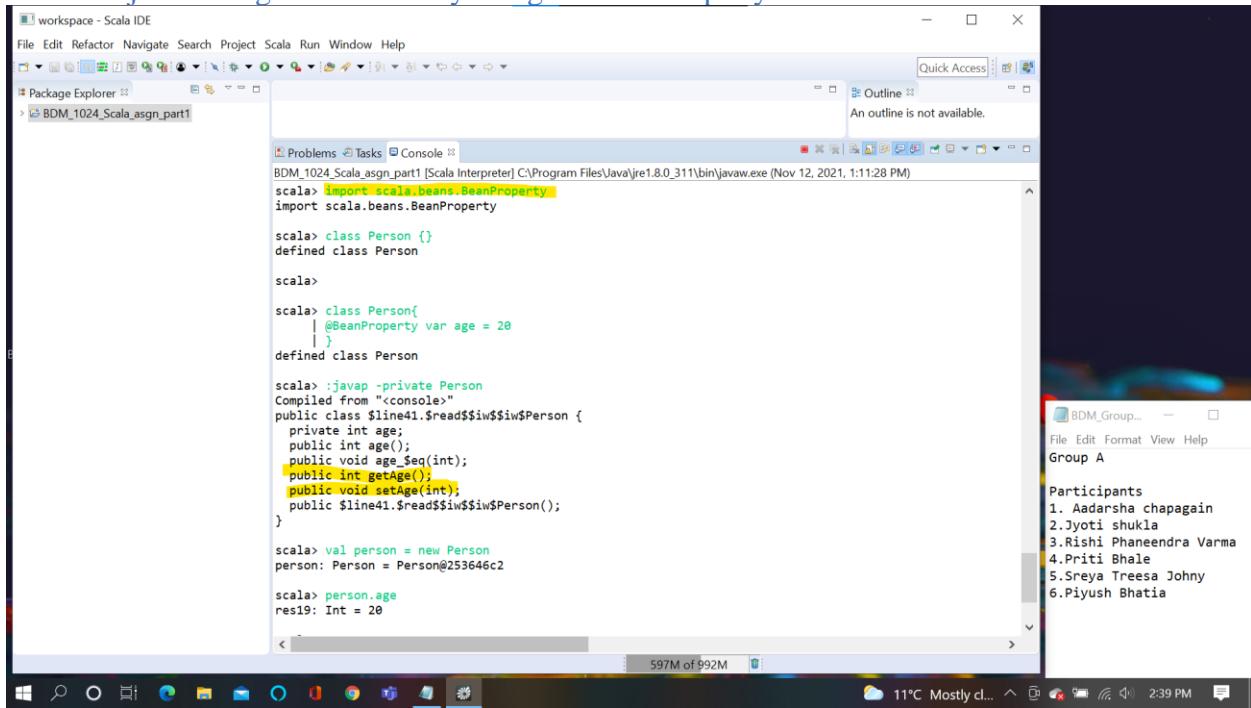
The screenshot shows the Eclipse IDE interface with the Scala IDE plugin. The console window displays the following Scala code and its execution:

```
workspace - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Package Explorer BDM_1024_Scala_asgn_part1
Problems Tasks Console
BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)
scala> person.age_=(-100)
scala> person.age
res14: Int = 20
scala> person.age_=(21)
scala> person.age
res16: Int = 21
scala> class Person {
    var thisAge = 20
    def age_= thisAge
    def age_=(newAge: Int) {
        if (newAge > 0 && newAge <= 100) thisAge = newAge
    }
}
defined class Person
scala> person.age
res17: Int = 21
scala> val person = new Person
person: Person = Person@13a80b56
scala> person.age_
res18: Int = 20
scala>
```

The right side of the interface shows a participant list for "Group A".

Slide 8:

Generate java time get and setter by using @BeansProperty



The screenshot shows the Eclipse IDE interface with the Scala IDE plugin. The console window displays the following Scala code and its execution, demonstrating the generation of Java beans properties:

```
workspace - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Package Explorer BDM_1024_Scala_asgn_part1
Problems Tasks Console
BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)
scala> import scala.beans.BeanProperty
import scala.beans.BeanProperty
scala> class Person {}
defined class Person
scala>
scala> class Person{
    | @BeanProperty var age = 20
    |
}
defined class Person
scala> :javap -private Person
Compiled from "<console>" public class $line41.$read$$iw$$iw$Person { private int age; private int age(); public void age_<eq>(int); public int getAge(); public void setAge(int); public $line41.$read$$iw$$iw$Person(); }
scala> val person = new Person
person: Person = Person@253646c2
scala> person.age
res19: Int = 20
```

The right side of the interface shows a participant list for "Group A".

Slide 9:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Left Sidebar:** Package Explorer (BDM_1024_Scala_asgn_part1).
- Console View:** Displays Scala code being typed and its output. The code defines a class Person with a private age field set to 20.
- Outline View:** An outline is not available.
- RCP View (Bottom Right):** Titled "BDM_Group..." with "Group A". It lists participants:
 - Participants
 - 1. Aadarsha chapagain
 - 2.Jyoti shukla
 - 3.Rishi Phaneendra Varma
 - 4.Priti Bhale
 - 5.Sreya Treesa Johny
 - 6.Piyush Bhatia
- System Tray:** Shows battery level (11°C Mostly cl...), signal strength, and time (2:43 PM).

Slide 10:

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Left Sidebar:** Package Explorer (BDM_1024_Scala_asgn_part1).
- Console View:** Displays Scala code defining a class Person with a private count field, increment method, and getCount method. It then creates an instance p1 and increments its count twice, resulting in a final value of 11.
- Outline View:** An outline is not available.
- RCP View (Bottom Right):** Titled "BDM_Group..." with "Group A". It lists participants:
 - Participants
 - 1. Aadarsha chapagain
 - 2.Jyoti shukla
 - 3.Rishi Phaneendra Varma
 - 4.Priti Bhale
 - 5.Sreya Treesa Johny
 - 6.Piyush Bhatia
- System Tray:** Shows battery level (11°C Mostly cl...), signal strength, and time (2:50 PM).

Slide 11:

Even if a particular attribute or variable in a class is private instances of the same class can refer to private variable directly because they are part of the class. Only when there is an object of another class then they are not able to refer to

The screenshot shows a Windows desktop with two instances of the Scala IDE running side-by-side. Both instances have the following Scala code in their consoles:

```
scala> class Person {
    private var count = 0

    def getCount() = count

    def increment(value: Int) {
        count += value
    }

    def compare(other: Person): Boolean = {
        (other.count >= count)
    }
}

scala> val p1 = new Person
p1: Person = Person@ee92d4e

scala> p1.increment(18)

scala> val p2 = new Person
p2: Person = Person@310d9735

scala> p2.increment(10)

scala> p1.getCount()
res29: Int = 18

scala> p2.getCount()
res30: Int = 10

scala> p1.compare(p2)
res31: Boolean = false

scala>
```

The code defines a `Person` class with a private attribute `count`. It includes methods to get the count and increment it. A `compare` method checks if one person's count is greater than or equal to another's. When two `Person` objects, `p1` and `p2`, are created and their counts are incremented, the `compare` method correctly returns `false` because the counts are different (18 and 10 respectively). This demonstrates that instances of the same class can directly access each other's private variables.

Slide 12:

private[this] var count “ means only available in the specific object either same class or different class

The screenshot shows the Eclipse IDE interface. In the center, there is a Scala code editor with the following content:

```
def getCount() = count

def increment(value: Int) {
    count += value
}

def compare(other: Person): Boolean = {
    (other.count >= count)
}

class Person {
    private[this] var count = 0

    def getCount() = count

    def increment(value: Int) {
        count += value
    }

    def compare(other: Person): Boolean = {
        (other.count >= count)
    }
}
```

In the terminal window (Console tab), the following command was run:

```
scala> class Person{
      private[this] var count = 0
    }
```

An error message is displayed in yellow:

```
<console>:22: error: value count is not a member of Person
          (other.count >= count)
```

Slide 13:

Here the primary constructor is not specified so it takes the default constructor.

The screenshot shows the Eclipse IDE interface. In the center, there is a Scala code editor with the following content:

```
class Person{
    private var age = 20
}
```

In the terminal window (Console tab), the following commands were run:

```
scala> :javap -private Person
Compiled from "<console>"
public class $line66.$read$$iw$$iw$Person {
    private int age;
    private int age();
    private void age_$eq(int);
    public $line66.$read$$iw$$iw$Person();
}

scala> val person = new Person
person: Person@2fd5f26

scala> class Person(var age: Int){}
defined class Person

scala> :javap -private Person
Compiled from "<console>"
public class $line68.$read$$iw$$iw$Person {
    private int age;
    public int age();
    public void age_$eq(int);
    public $line68.$read$$iw$$iw$Person(int);
}

scala> val person = new Person
<
```

Slide 14:

Here we have specified the constructor so while we initialize without any argument it throws error.

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** workspace - Scala IDE, File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Left Sidebar:** Package Explorer (BDM_1024_Scala_asgn_part1).
- Center Area:** Problems, Tasks, Console tab selected. It shows Scala code and Java output:

```
BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)
Compiled from "<console>"
```

```
public class $line66.$read$$iw$$iw$Person {
    private int age;
    private int age();
    private void age_$eq(int);
    public $line66.$read$$iw$$iw$Person();
}
```

```
scala> val person = new Person
person: Person@2fd5f26
```

```
scala> class Person(var age: Int){}
defined class Person
```

```
scala> :javap -private Person
Compiled from "<console>"
```

```
public class $line68.$read$$iw$$iw$Person {
    private int age;
    public int age();
    public void age_$eq(int);
    public $line68.$read$$iw$$iw$Person(int);
}
```

```
scala> val person = new Person
<console>:13: error: not enough arguments for constructor Person: (age: Int)Person.
Unspecified value parameter age.
          val person = new Person
                           ^

```

```
scala>
```
- Right Side:** Outline (An outline is not available), Java application window titled "BDM_Group..." showing "Group A" participants:

Participants
1. Aadarsha chapagain
2. Jyoti shukla
3. Rishi Phaneendra Varma
4. Priti Bhale
5. Sreya Treesa Johnny
6. Piyush Bhatia
- Bottom:** Taskbar showing system status (11°C, 3:22 PM).

Slide 15:

Passing the argument(20 in this case) works. Since that is the argument the class is asking.

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** workspace - Scala IDE, File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Left Sidebar:** Package Explorer (BDM_1024_Scala_asgn_part1).
- Center Area:** Problems, Tasks, Console tab selected. It shows Scala code and Java output:

```
BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)
scala> class Person(var age: Int){}
defined class Person
```

```
scala> :javap -private Person
Compiled from "<console>"
```

```
public class $line68.$read$$iw$$iw$Person {
    private int age;
    public int age();
    public void age_$eq(int);
    public $line68.$read$$iw$$iw$Person(int);
}
```

```
scala> val person = new Person
<console>:13: error: not enough arguments for constructor Person: (age: Int)Person.
Unspecified value parameter age.
          val person = new Person
                           ^

```

```
scala> val person = new Person(20)
person: Person@549917cf
```

```
scala> person.age
res32: Int = 20
```

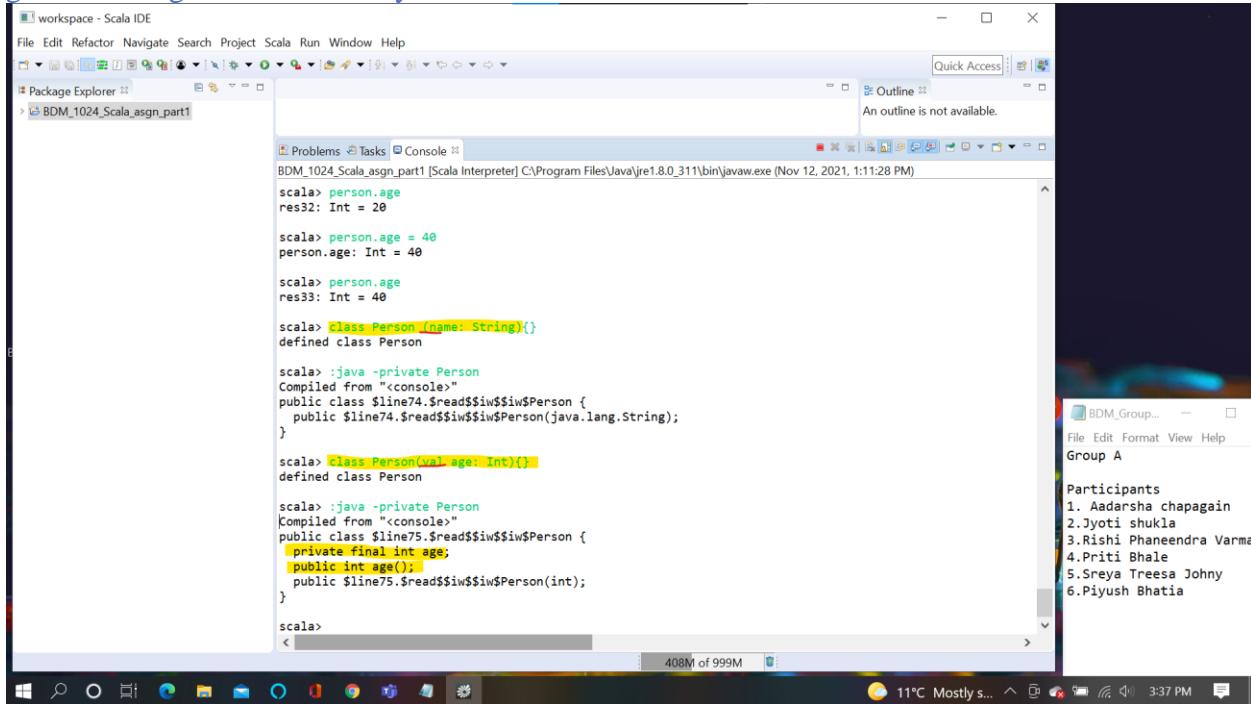
```
scala> person.age = 40
person.age: Int = 40
```

```
scala> person.age
res33: Int = 40
```
- Right Side:** Outline (An outline is not available), Java application window titled "BDM_Group..." showing "Group A" participants:

Participants
1. Aadarsha chapagain
2. Jyoti shukla
3. Rishi Phaneendra Varma
4. Priti Bhale
5. Sreya Treesa Johnny
6. Piyush Bhatia
- Bottom:** Taskbar showing system status (11°C, 3:28 PM).

Slide 16 and 17:

If val is not specified it will not create any getter or setter method. If val is specified it will generate the getter method only.



The screenshot shows the Eclipse IDE interface with the Scala IDE plugin. The 'Console' tab is active, displaying Scala code and its corresponding Java output. The code defines a class Person with an age field. In the first part, 'age' is defined as a regular field (not val), so both getter and setter methods are generated. In the second part, 'age' is defined as a val, which generates only the getter method. The Java output shows the generated code for both cases.

```
workspace - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Package Explorer BDM_1024_Scala_asgn_part1
Problems Tasks Console
BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)
scala> person.age
res32: Int = 28

scala> person.age = 40
person.age: Int = 40

scala> person.age
res33: Int = 40

scala> class Person (name: String){}
defined class Person

scala> :java -private Person
Compiled from "<console>""
public class $line74.$read$$iw$$iw$Person {
    public $line74.$read$$iw$$iw$Person(java.lang.String);
}

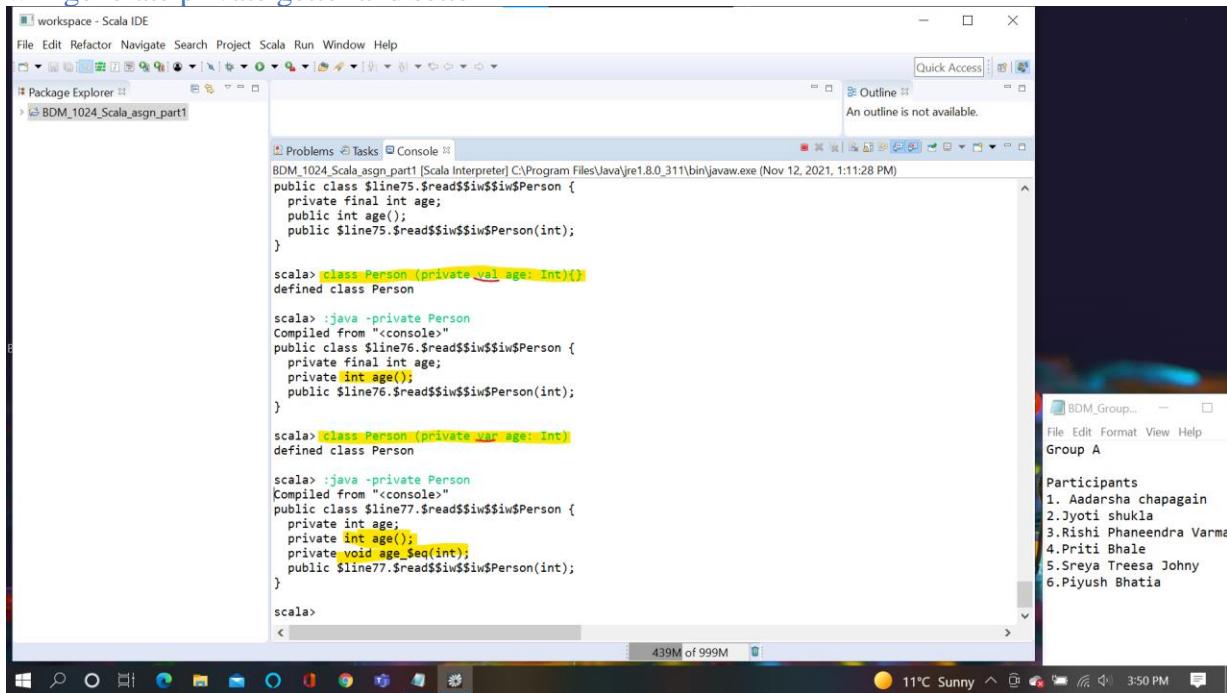
scala> class Person(val age: Int){}
defined class Person

scala> :java -private Person
Compiled from "<console>""
public class $line75.$read$$iw$$iw$Person {
    private final int age;
    public int age();
    public $line75.$read$$iw$$iw$Person(int);
}

scala>
```

Slide 18:

When you define it as private.If it is val it will generates private getter but not setter.If it is var it will generate private getter and setter



The screenshot shows the Eclipse IDE interface with the Scala IDE plugin. The 'Console' tab is active, displaying Scala code and its corresponding Java output. The code defines a class Person with an age field. In the first part, 'age' is defined as a private val, which generates only a private getter method. In the second part, 'age' is defined as a private var, which generates both a private getter and a private setter method. The Java output shows the generated code for both cases.

```
workspace - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Package Explorer BDM_1024_Scala_asgn_part1
Problems Tasks Console
BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)
public class $line75.$read$$iw$$iw$Person {
    private final int age;
    public int age();
    public $line75.$read$$iw$$iw$Person(int);
}

scala> class Person (private val age: Int){}
defined class Person

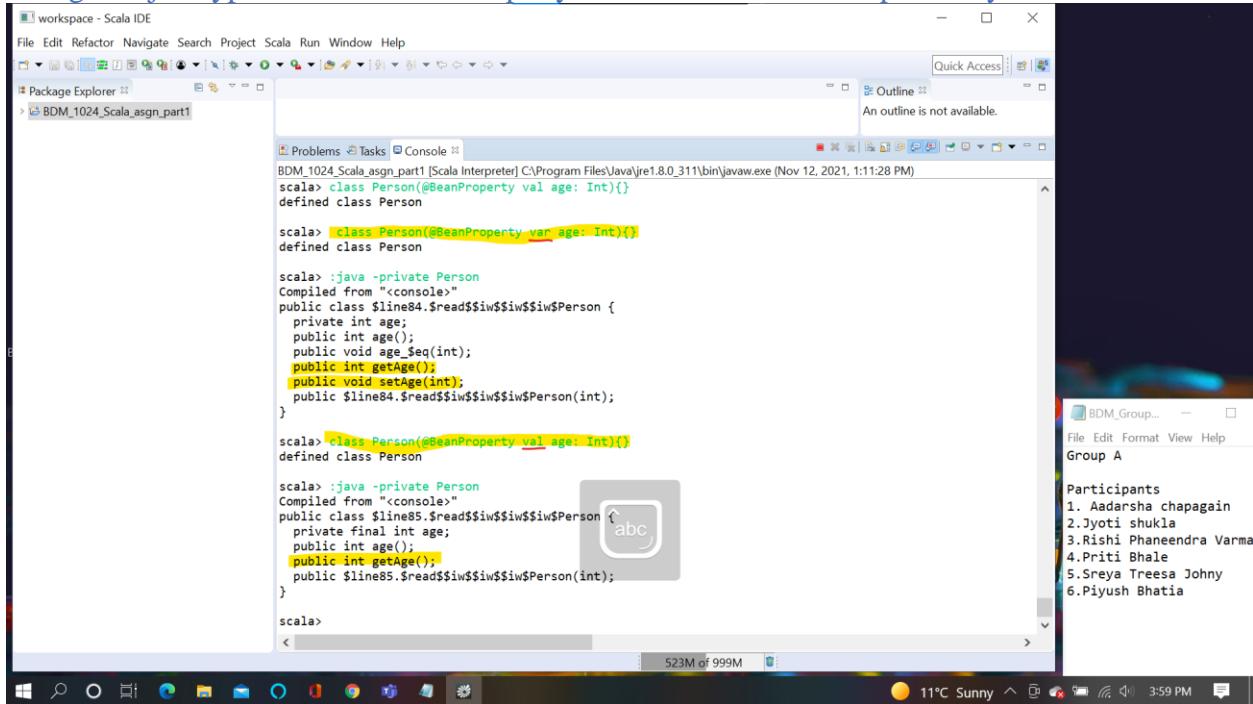
scala> :java -private Person
Compiled from "<console>""
public class $line76.$read$$iw$$iw$Person {
    private final int age;
    private int age();
    public $line76.$read$$iw$$iw$Person(int);
}

scala> class Person (private var age: Int)
defined class Person

scala> :java -private Person
Compiled from "<console>""
public class $line77.$read$$iw$$iw$Person {
    private int age;
    private int age();
    private void age_$eq(int);
    public $line77.$read$$iw$$iw$Person(int);
}
```

Slide 19:

When `@BeanProperty` is used it will generate java getter and setter for var
And getter javatype for val. `@BeanProperty` is meant for Java interoperability.



```
workspace - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Package Explorer BDM_1024_Scala_asgn_part1
Problems Tasks Console
BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)
scala> class Person(@BeanProperty val age: Int){}
defined class Person

scala> class Person(@BeanProperty var age: Int){}
defined class Person

scala> :java -private Person
Compiled from "<console>"
public class $line84.$read$$iw$$iw$$iw$Person {
    private int age;
    public int age();
    public void age_$eq(int);
    public int getAge();
    public void setAge(int);
    public $line84.$read$$iw$$iw$$iw$Person(int);
}

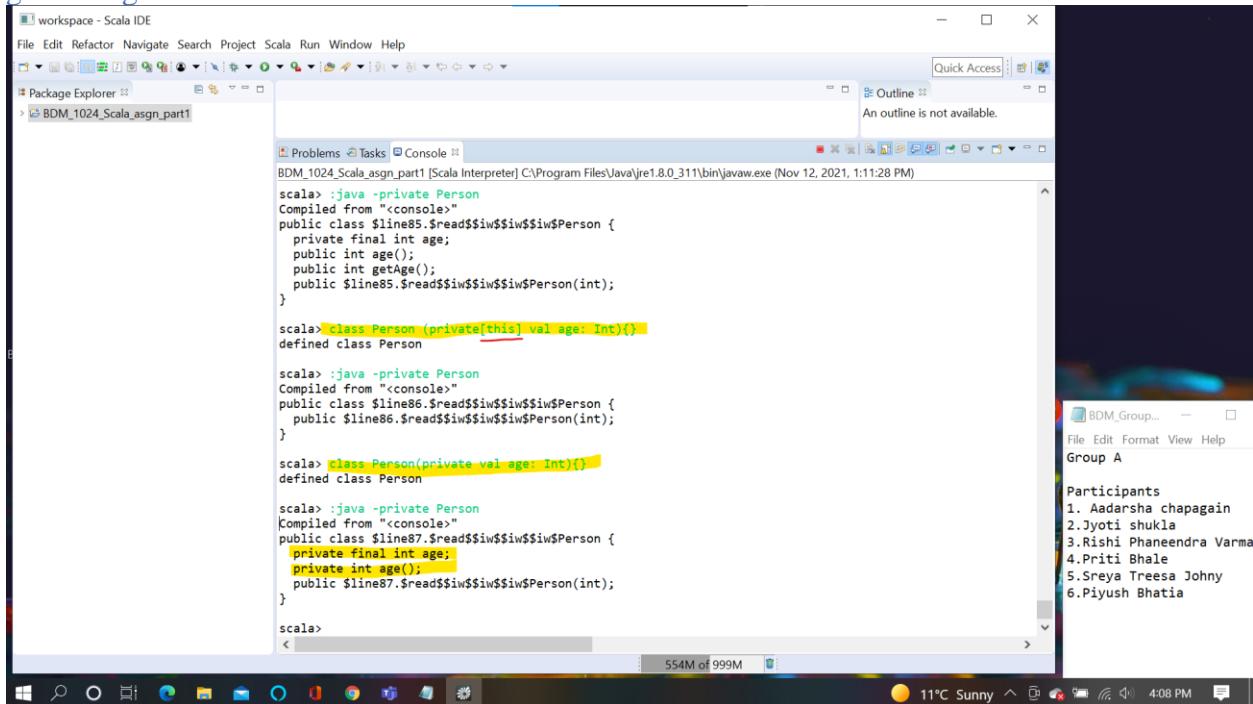
scala> class Person(@BeanProperty val age: Int){}
defined class Person

scala> :java -private Person
Compiled from "<console>"
public class $line85.$read$$iw$$iw$$iw$Person {
    private final int age;
    public int age();
    public int getAge();
    public $line85.$read$$iw$$iw$$iw$Person(int);
}

scala>
```

Slide 20:

When we use private [this] it will be only available with that object that's why it does not generate getter and setter



```
workspace - Scala IDE
File Edit Refactor Navigate Search Project Scala Run Window Help
Package Explorer BDM_1024_Scala_asgn_part1
Problems Tasks Console
BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 1:11:28 PM)
scala> :java -private Person
Compiled from "<console>"
public class $line85.$read$$iw$$iw$$iw$Person {
    private final int age;
    public int age();
    public int getAge();
    public $line85.$read$$iw$$iw$$iw$Person(int);
}

scala> class Person (private[this] val age: Int){}
defined class Person

scala> :java -private Person
Compiled from "<console>"
public class $line86.$read$$iw$$iw$$iw$Person {
    public $line86.$read$$iw$$iw$$iw$Person(int);
}

scala> class Person(private val age: Int){}
defined class Person

scala> :java -private Person
Compiled from "<console>"
public class $line87.$read$$iw$$iw$$iw$Person {
    private final int age;
    private int age();
    public $line87.$read$$iw$$iw$$iw$Person(int);
}

scala>
```

Slide 21:

In both classes variable "age" is defined but in the second one it is used within procedure however in the first one it is not used anywhere.

In the second one "age" is used in the procedure so, Variable become a class variable
If val or var is not used and the if the variable is not used anywhere it cannot be a class variable

The screenshot shows the Eclipse IDE interface with the following details:

- File Explorer:** Shows a project named "BDM_1024_Scala_asgn_part1".
- Console:** Displays Scala code being typed into the interpreter. The code defines two classes: "Person" with an Int parameter and "Person" with an Int variable. It also creates an instance of the second "Person" class and prints its age.
- Participants:** A list of six names: Aadarsha chapagain, Jyoti shukla, Rishi Phaneendra Varma, Priti Bhale, Sreya Treesa Johny, and Piyush Bhatia.

Slide 22:

Anything that follows the definition of primary constructor within a class becomes a code which get executed when primary constructor executed

The screenshot shows the Eclipse IDE interface with the following details:

- File Explorer:** Shows a project named "BDM_1024_Scala_asgn_part1".
- Console:** Displays Scala code defining a class "Person" with a primary constructor taking an Int parameter. Inside the class, there is a println statement and a showAge method. An instance of the class is created, and its showAge method is called, outputting the value 20.
- Participants:** A list of six names: Aadarsha chapagain, Jyoti shukla, Rishi Phaneendra Varma, Priti Bhale, Sreya Treesa Johny, and Piyush Bhatia.

Slide 23:

The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Toolbars:** Standard toolbar, Quick Access toolbar.
- Views:** Package Explorer (BDM_1024_Scala_asgn_part1), Outline (An outline is not available).
- Console:** Displays Scala code execution:

```
scala> :java -private Person
Compiled from "c$console"
public class $line94.$read$$iw$$iw$$iw$Person {
    private int age;
    public int age();
    public void age_$eq(int);
    public void showAge();
    public $line94.$read$$iw$$iw$Person(int);
}

scala> person.showAge
20

scala> class Person(var age: Int){
    |   println("Executing Primary constructor Line1")
    |   println("Executing Primary constructor Line2")
    |
    |   def showAge(){
    |       println(age)
    |   }
    | }
defined class Person

scala> val person = new Person(20)
Executing Primary constructor Line1
Executing Primary constructor Line2
person: Person@78f9b4a1

scala>
```
- Participants:** Group A (Aadarsha chapagain, Jyoti shukla, Rishi Phaneendra Varma, Priti Bhale, Sreya Treesa Johny, Piyush Bhatia).
- System Status:** 10°C Mostly cl..., 4:32 PM.

58 ,61,62,63 ,72,73 ,74,76

Slide 58:

The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Toolbars:** Standard toolbar, Quick Access toolbar.
- Views:** Package Explorer (BDM_1024_Scala_asgn_part1), Outline (ScalaTraining, Main).
- Console:** Displays Scala code execution:

```
scala> package ScalaTraining
package ScalaTraining

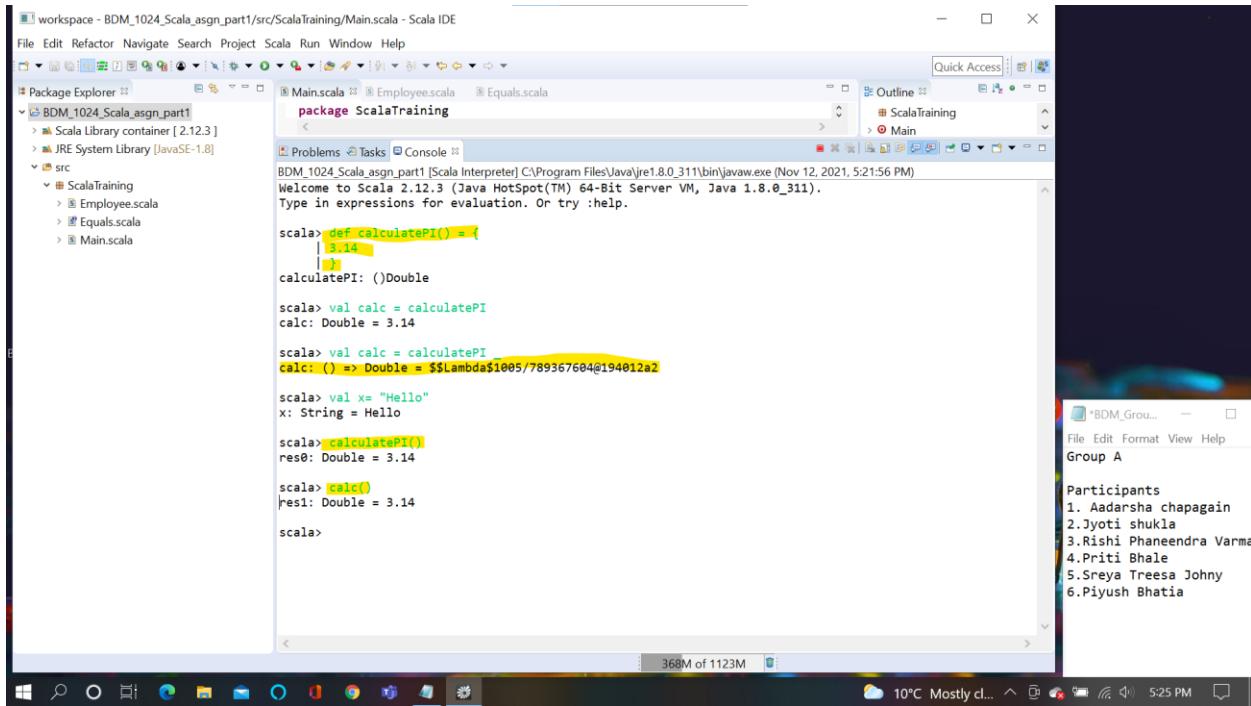
scala> object Main {
    |   def main (args: Array[String]){
    |       val emp1 = new Employee(500, "Peter")
    |       val emp2 = new Employee(501, "Parker")
    |
    |       println("Are the 2 employee the same:" + emp1.isEqual(emp2))
    |       println("Emp1 salary in Double:" + emp1.convert.ToDouble(50))
    |       println("Emp2 salary in Double:" + emp1.convert.ToDouble(2500))
    |   }
}
```

```
<terminated> Main$ [Scala Application] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 5:15:13 PM)
Are the 2 employee the same:false
Emp1 salary in Double:50.0
Invalid Number.
Emp2 salary in Double:0.0
```
- Participants:** Group A (Aadarsha chapagain, Jyoti shukla, Rishi Phaneendra Varma, Priti Bhale, Sreya Treesa Johny, Piyush Bhatia).
- System Status:** 10°C Mostly cl..., 5:15 PM.

Slide61:

Calc and x are both variable but calc is pointed to a function. calc holds a function .

Look at the type of variable is “`() => Double = <function0>`” `()` means unit `.()` : this is a function which takes unit and returns a double.(calculatePI takes a unit and returns double)



```
workspace - BDM_1024_Scala_asgn_part1/src/ScalaTraining/Main.scala - Scala IDE
File Edit Refactor Navigate Project Scala Run Window Help
Package Explorer Problems Tasks Console
BDM_1024_Scala_asgn_part1 [ 2.12.3 ] JRE System Library [ JavaSE-1.8 ]
src ScalaTraining Employee.scala Equals.scala Main.scala
Main.scala Employee.scala Equals.scala Main.scala
package ScalaTraining
def calculatePI() = {
    3.14
}
calculatePI: ()Double

val calc = calculatePI
calc: Double = 3.14

val calc = calculatePI_
calc: () => Double = $$Lambda$1005/789367604@194012a2

val x = "Hello"
x: String = Hello

calculatePI_()
res0: Double = 3.14

calc()
res1: Double = 3.14

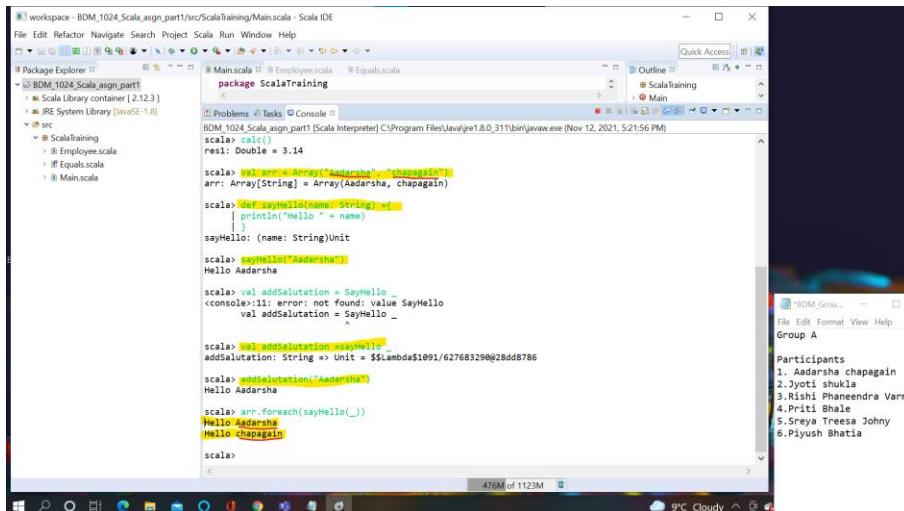
scala>
```

Slide62:

_ means current value in another word. it is basically designate the current element variable

.Foreach is higher order function which accept another function

Function is supposed to take a variable but here foreach which is a function is taking another function



```
workspace - BDM_1024_Scala_asgn_part1/src/ScalaTraining/Main.scala - Scala IDE
File Edit Refactor Navigate Project Scala Run Window Help
Package Explorer Problems Tasks Console
BDM_1024_Scala_asgn_part1 [ 2.12.3 ] JRE System Library [ JavaSE-1.8 ]
src ScalaTraining Employee.scala Equals.scala Main.scala
Main.scala Employee.scala Equals.scala Main.scala
package ScalaTraining
def calculatePI() = {
    3.14
}
calculatePI: ()Double

val arr = Array("Aadarsha", "chapagain")
arr: Array[String] = Array(Aadarsha, chapagain)

def sayHello(name: String) = {
    println("Hello " + name)
}

sayHello: (name: String)Unit

sayHello("Aadarsha")
Hello Aadarsha

val addSalutation = SayHello_
<console>:11: error: not found: value SayHello
      val addSalutation = SayHello_
                           ^
addSalutation: String => Unit = $$Lambda$1091/627683290028dd8786

val addSalutation: String => Unit = $$Lambda$1091/627683290028dd8786

addSalutation("Aadarsha")
Hello Aadarsha

arr.foreach(sayHello(_))
Hello Aadarsha
Hello chapagain
Hello chapagain

scala>
```

Slide63:

Map is another higher order function

The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Toolbars:** Standard, Selection, Status Bar.
- Views:** Package Explorer, Main.scala, Employee.scala, Equals.scala, Problems, Tasks, Console, Outline, Quick Access.
- Console Output:** BDM_1024_Scala_asgn_part1 [Scala Interpreter] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 5:21:56 PM)
scala> val addSalutation =sayHello_
addSalutation: String => Unit = \$\$Lambda\$1091/627683290@28dd8786
scala> addSalutation("Aadarsha")
Hello Aadarsha
scala> arr.foreach(sayHello(_))
Hello Aadarsha
Hello chapagain
scala> def double (x: Int)={
double: (x: Int)Unit
scala> def double (x: Int)={
| x*2
| }
double: (x: Int)Int
scala> double(3)
res5: Int = 6
scala> val arr = Array(2, 50, 28, 36)
arr: Array[Int] = Array(2, 50, 28, 36)
scala> val newArr = arr.map(double(_))
newArr: Array[Int] = Array(4, 100, 56, 72)
scala>
- Participants:** Group A (Aadarsha chapagain, Jyoti shukla, Rishi Phaneendra Varma, Priti Bhale, Sreya Treessa Johnny, Piyush Bhatia).

Slide72:

The screenshot shows the Eclipse IDE interface with the following details:

- File Menu:** File, Edit, Refactor, Navigate, Search, Project, Scala, Run, Window, Help.
- Toolbars:** Standard, Selection, Status Bar.
- Views:** Package Explorer, Main.scala, Employee.scala, Equals.scala, HOFDemo1.scala, Problems, Tasks, Console, Outline, Quick Access.
- Console Output:** workspace - BDM_1024_Scala_asgn_part1/src/ScalaTraining/HOFDemo1.scala - Scala IDE
BDM_1024_Scala_asgn_part1 [Scala Application] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Nov 12, 2021, 5:52:28 PM)
18
28
36
46
- Participants:** Group A (Aadarsha chapagain, Jyoti shukla, Rishi Phaneendra Varma, Priti Bhale, Sreya Treessa Johnny, Piyush Bhatia).

Slide73:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "BDM_1024_Scala_asgn_part1/src/ScalaTraining".
- Code Editor:** Displays the file "HOFDemo2.scala" containing Scala code. The code defines a main function that takes an array of strings and performs an operation on each element. The operation is defined by a function "performOp" which takes an integer and returns a string.
- Console:** Shows the output of the Scala application running in the Java environment. The output is:

```
10
20
30
40
```
- Participants:** A sidebar titled "Participants" lists six names: 1. Aadarsha chapagain, 2. Jyoti shukla, 3. Rishi Phaneendra Varma, 4. Priti Bhale, 5. Sreya Treesa Johny, and 6. Piyush Bhatia.

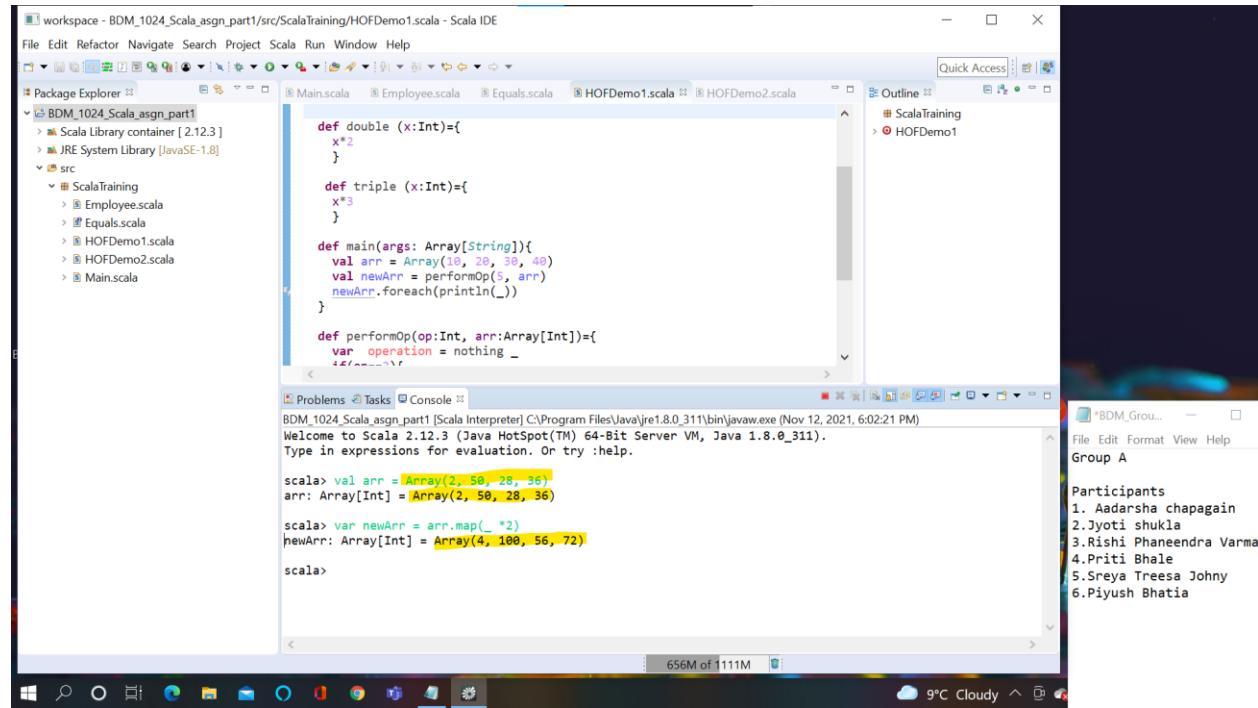
Slide74:

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows the project structure under "BDM_1024_Scala_asgn_part1/src/ScalaTraining".
- Code Editor:** Displays the file "HOFDemo2.scala" containing Scala code. The code is identical to the one in Slide 73, defining a main function and a performOp function.
- Console:** Shows the output of the Scala application running in the Java environment. The output is:

```
10
20
30
40
50
60
70
80
```
- Participants:** A sidebar titled "Participants" lists six names: 1. Aadarsha chapagain, 2. Jyoti shukla, 3. Rishi Phaneendra Varma, 4. Priti Bhale, 5. Sreya Treesa Johny, and 6. Piyush Bhatia.

Slide76:



Part 2 :Write a Scala code which can get data from Twitter and show in the console.
Please upload your code too

```
object FetchTwitterData {
    import sttp.client3.quick._
    def main(args: Array[String]){
        val BearerToken = "<tokenfromtwitter>"
        val UserName = "<twitterUserName>"
        val TwitterEndpoint = "https://api.twitter.com/2/users/by/username/$UserName"
        val GetResponse = basicRequest
        .get(TwitterEndpoint)
        .auth
        .bearer(BearerToken)
        println.GetResponse
    }
}
```