

COMP 250

INTRODUCTION TO COMPUTER SCIENCE

Week 3-2: OOD: Packages, Fields, and Modifiers

Giulia Alberini, Fall 2020

WHAT ARE WE GOING TO DO IN THIS VIDEO?



OOD1

- Packages
- Fields
- Modifiers

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

WARNING!



Assignment Project Exam Help

- This is a terminology-heavy lecture!

<https://powcoder.com>

- Do not hesitate to stop the video and
rewind.

Add WeChat powcoder

Assignment Project Exam Help

PACKAGES

<https://powcoder.com>

Add WeChat powcoder

PACKAGES

- A **package** is a group of classes
 - Each class is referred to as a *package member*
- A **class** is a group of methods
- A **method** is an ordered group of commands

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

DEFINITION

- To define a package we write at the top of our class file the following statement

`package packageName;`

- For example:

<https://powcoder.com>

```
package nba.annoyingTeams;  
  
public class MiamiHeat {  
    ⋮  
}
```

This creates a class `MiamiHeat` inside the package `nba.annoyingTeams`

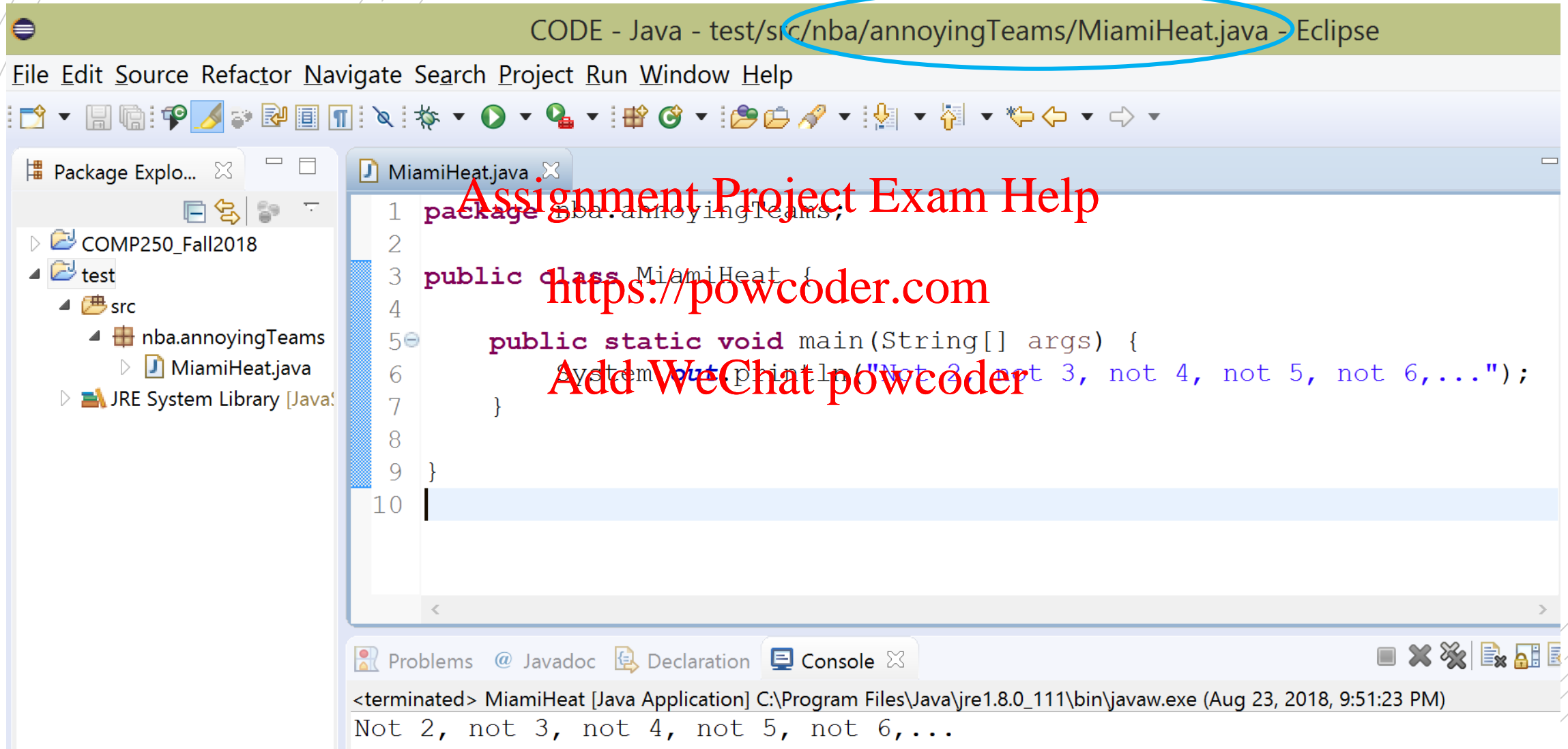
FILE AND FOLDERS NAMES

There are two main rules related to files' and folders' names in Java:

1. The name of the *class* must match the name of the file (with .java added) (e.g. *MiamiHeat.java*) <https://powcoder.com>
2. The folder path must match exactly the package name – except that each period is actually a "slash" (i.e. a subfolder)

In the example before, a folder *nba* must contain a folder *annoyingTeams* which contains the file *MiamiHeat.java*

EXAMPLES



CODE - Java - test/src/nba/annoyingTeams/MiamiHeat.java - Eclipse

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer: COMP250_Fall2018 > test > src > nba.annoyingTeams > MiamiHeat.java

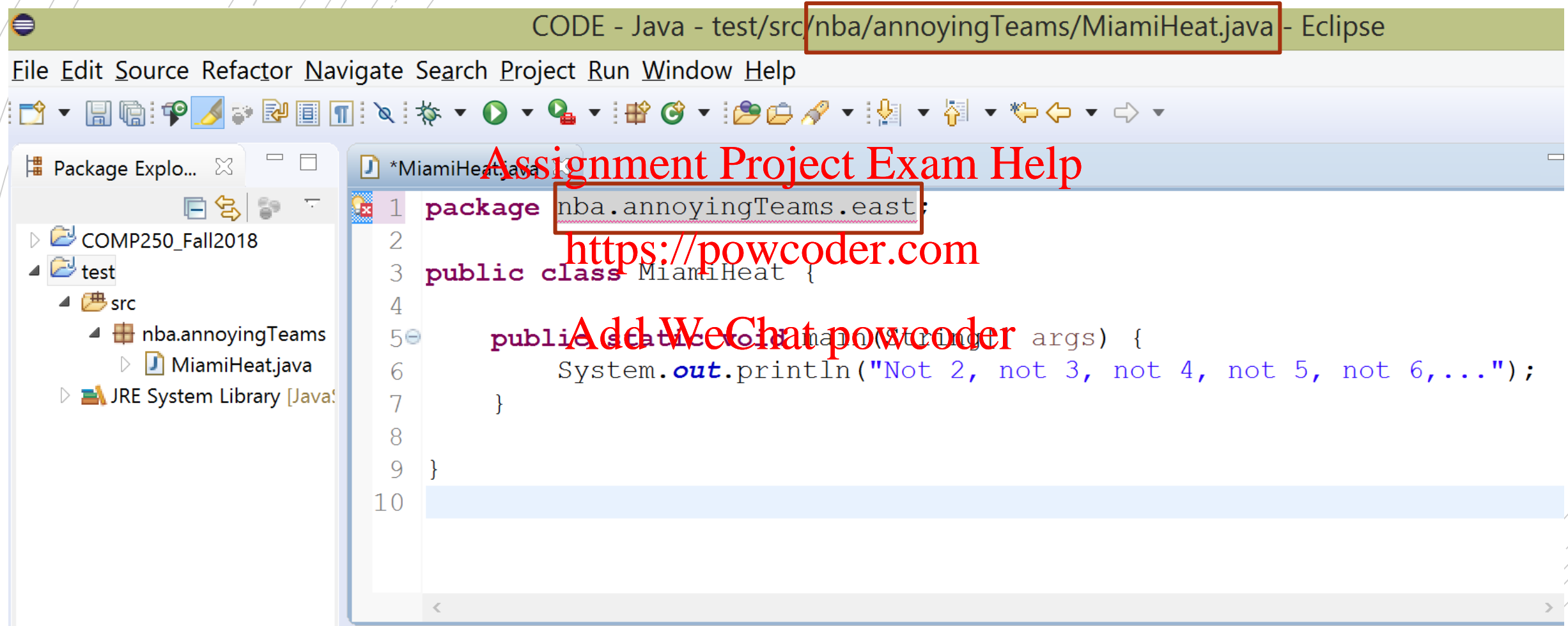
```
1 package nba.annoyingTeams;
2
3 public class MiamiHeat {
4     public static void main(String[] args) {
5         System.out.println("Not 2, not 3, not 4, not 5, not 6,...");
6     }
7 }
8
9
10
```

Problems Javadoc Declaration Console

<terminated> MiamiHeat [Java Application] C:\Program Files\Java\jre1.8.0_111\bin\javaw.exe (Aug 23, 2018, 9:51:23 PM)

Not 2, not 3, not 4, not 5, not 6,...

EXAMPLES



CODE - Java - test/src/nba/annoyingTeams/MiamiHeat.java - Eclipse

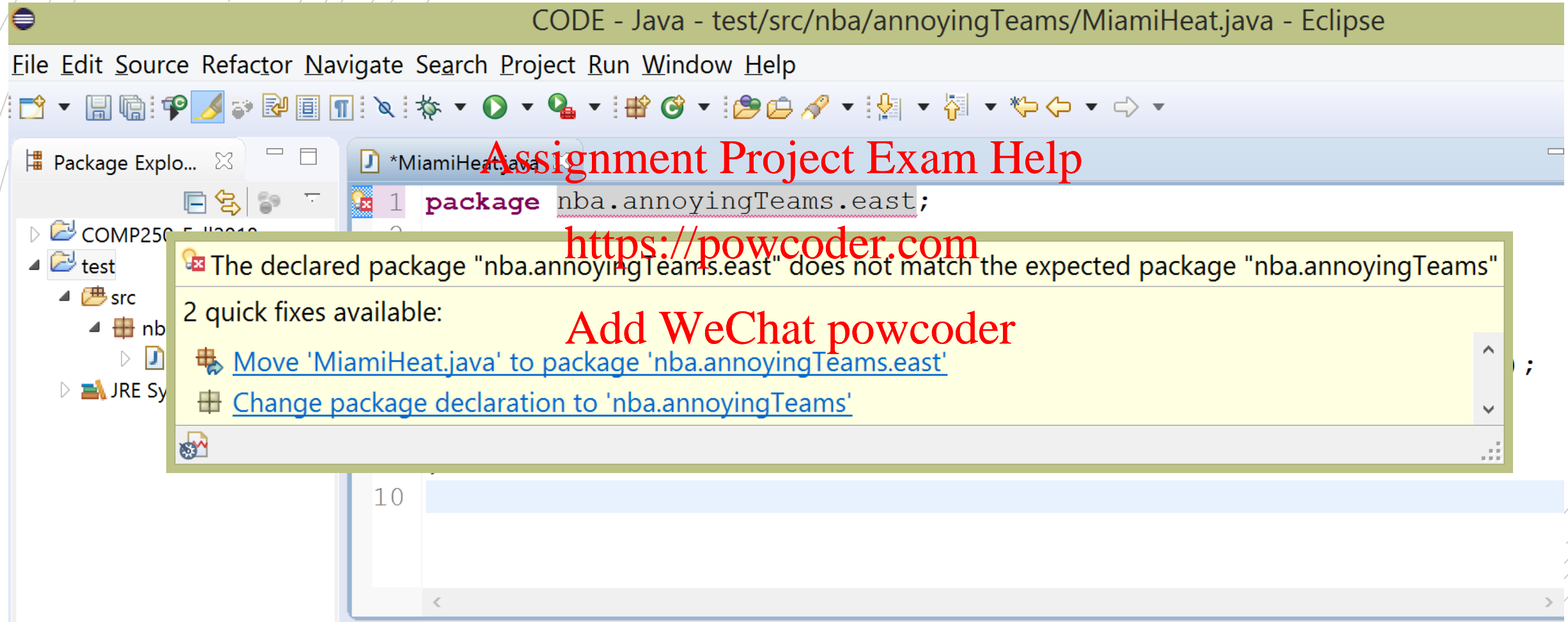
File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- COMP250_Fall2018
 - test
 - src
 - nba.annoyingTeams
 - MiamiHeat.java
- JRE System Library [JavaSE-7]

```
1 package nba.annoyingTeams.east;
2
3 public class MiamiHeat {
4
5     public static void main(String[] args) {
6         System.out.println("Not 2, not 3, not 4, not 5, not 6,...");
7     }
8
9 }
10
```

EXAMPLES



PACKAGES

java.lang

Object.java

String.java

Math.java

System.java

java.util

Random.java

Arrays.java

ArrayList.java

nba.annoyingTeams

MiamiHeat.java

animals

Cat.java

Dog.java

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

USING A CLASS IN YOUR PROGRAM

If you want to use a *package member* from *outside* its package, you must instruct your program where to find that class. You can do this in 3 ways:

1. Specify the entire path whenever you use such class.

For example, whenever you want to use `Dog` from the `animals` package you can *fully qualify* the class name: `animals.Dog`

```
animals.Dog myDog = new animals.Dog();
```

Ok for infrequent use!

USING A CLASS IN YOUR PROGRAM

If you want to use a *package member* from *outside* its package, you must instruct your program where to find that class. You can do this in 3 ways:

2. Import the package member. Example:

<https://powcoder.com>
Add WeChat powcoder

```
import animals.Dog;
```

This tells the computer that the class Dog is found in the package animals.

Ok if you use few members
from a package.

USING A CLASS IN YOUR PROGRAM

If you want to use a *package member* from *outside* its package, you must instruct your program where to find that class. You can do this in 3 ways:

3. Import the entire package. Example:

```
import animals.*;
```

Now you can refer to any class inside the `animals` package.

USING A CLASS IN YOUR PROGRAM

For convenience, the Java compiler automatically imports two entire packages for each source file:

Assignment Project Exam Help

1. The `java.lang` package <https://powcoder.com>

2. The *current* package Add WeChat powcoder

This is why no import statement is need to use `Math`, `String`, ... , or any *package member* from **inside** its own package.

Assignment Project Exam Help

INTRO TO OOP

<https://powcoder.com>

Add WeChat powcoder

RANDOM

How do you use it?

1. Import the corresponding class:

- Identify the class in which we want to use it
- Before the class definition, add the following statement:

```
import java.util.Random;
```

RANDOM

How do you use it?

2. Declare a variable of type `Random`, and create the object using the operator `new`.

<https://powcoder.com>

Add WeChat powcoder

```
Random randomGenerator = new Random();  
Random otherGenerator = new Random(seed);
```

RANDOM

How do you use it?

2. Declare a variable of type `Random`, and create the object using the operator `new`.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

```
Random randomGenerator = new Random();  
Random otherGenerator = new Random(seed);
```

Declaration of two
variables of type `Random`.

RANDOM

How do you use it?

2. Declare a variable of type `Random`, and create the object using the operator `new`.

<https://powcoder.com>

Add WeChat powcoder

```
Random randomGenerator = new Random();  
Random otherGenerator = new Random(seed);
```

Declaration of two
variables of type `Random`.

Creation of two `Random` **objects**.
Note: the result of the `new` operator is
a ***reference*** to the new object.

RANDOM

How do you use it?

3. We called methods on the objects we created using the dot (.) operator

<https://powcoder.com>

Add WeChat powcoder

```
Random randomGenerator = new Random();  
int randomNumber = randomGenerator.nextInt(100);
```

OBJECTS

Assignment Project Exam Help

and

<https://powcoder.com>

CLASSES

Add WeChat powcoder

OBJECTS

- An object is a collection of data and a set of methods can be provided to work with it. For example, a `String` is a collection of characters and methods like `charAt()` and `length()` can be used on it.
- Java is an object-oriented language. This means that it uses objects to represent data and provides methods related to them.
- Methods can take objects as parameters and produce objects as return values.
- Type of Objects we have seen up to now: `String`, `arrays`, `Random`.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

IDEA: DEFINE YOUR OWN TYPE

- In Java, we can define our own type of data.

Assignment Project Exam Help

- The idea is that we can combine related pieces of information with each other into one variable.

<https://powcoder.com>

Add WeChat powcoder

HOW TO DEFINE A NEW TYPE

- When you define a class, you are actually defining a new type.

Assignment Project Exam Help

- So after defining `HelloWorld` you actually defined a type `HelloWorld`.

<https://powcoder.com>

- This means that in another `.java` file, you actually could declare a variable of type `HelloWorld`.

Add WeChat powcoder

- This wouldn't really make sense though because the new type doesn't *store* anything.

UP TO NOW

- Up to now we have created and used classes as containers for static methods. These kind of classes are called **Utility classes**. An example of a utility class is the `Math` class.
<https://powcoder.com>
- However, Java is an Object Oriented Programming language. In java classes can have a much bigger role!

CLASSES

By now, we should all know that objects and classes are closely related.
How exactly?

Assignment Project Exam Help

- Each time we define a **class** we create a new **object type** with the same name.

<https://powcoder.com>

Add WeChat powcoder

- A class is a blueprint/template for a type of object. It specifies what properties the objects have and what methods can operate on them.
- An object is an **instance** of some class.

THE BLUEPRINT

```
public class ClassName {
```

```
    // some data declared here
```

```
    <modifier> <type> <variable_name>;
```

```
    public ClassName() {
```

```
        //constructor
```

```
    }
```

```
    // declare other methods
```

```
}
```

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Data

Method to create an
object

Other methods

File name: **ClassName.java**

NOTE ON NESTED CLASSES

- You can define a class *within* another class. We call such class a *nested class*. We refer to the class containing a nested class as the *outer class*.

Assignment Project Exam Help

- Why?

<https://powcoder.com>

- To group classes that are used only in one place.

Add WeChat powcoder

If a class is useful to only one class, it makes sense to keep it nested and together.

- Increase encapsulation.

Allows for better control over data.

- Create readable and maintainable code.

Assignment Project Exam Help

FIELDS

<https://powcoder.com>

Add WeChat powcoder

STEP 1

```
public class ClassName {
```

```
// some data declared here
```

```
<modifier> <type> <variable_name>;
```

Data

```
public ClassName () {
```

```
//constructor
```

```
}
```

<https://powcoder.com>

Add WeChat powcoder

Method to create an object

```
// declare other methods
```

Other methods

```
}
```

File name: **ClassName.java**

STEP 1 – DATA

- Variables that denote the data stored by an object are usually called **fields** (or **attributes**).

<https://powcoder.com>

- When defining a new data type the first thing to decide is the following:
 - What should be an attribute/field?
 - What type should these attributes/fields be?

SYNTAX

- fields are declared at the beginning of the class definition, outside of any method.

Assignment Project Exam Help

<https://powcoder.com>

- Syntax:

Add WeChat powcoder

```
<modifier> <type> <variable_name>;
```

```
<modifier> <type> <variable_name>;
```

Modifiers are **keyword** that you add to class/method/variable's definition to change their meaning. Java has different kind of modifiers, including:

Assignment Project Exam Help

<https://powcoder.com>

Access Control Modifiers

- **public**
- **protected**
- *default* (no keyword)
- **private**

Non-Access Modifiers

- **static**
- **final**
- **abstract**

Add WeChat powcoder

PRIVATE VS PUBLIC

- They are access control modifiers
(keywords that determine from where a method or a variable can be accessed)

- `private` **Assignment Project Exam Help**

The method or variable that comes after is only accessible within the class in which it was written.

<https://powcoder.com>

Add WeChat powcoder
`private int dontTouchMe;`

- `public`

The method or variable that comes after is accessible from anywhere

`public int lookAtMe;`

VISIBILITY/ACCESS CONTROL MODIFIERS

Note:

- outer classes can only be declared `public` or `package private`.
- members of a class (fields, methods, classes) can be declared using any of the access modifiers.

- `public`
- `protected` (= package + subclasses)
- `default` (= package)
- `private`

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

These modifiers define what is visible across classes.

Modifier	Class	Package	Subclass	World
<code>public</code>	Y	Y	Y	Y
<code>protected</code>	Y	Y	Y	N
<i><code>no modifier</code></i>	Y	Y	Y/N	N
<code>private</code>	Y	N	N	N

DEMO

1. Create a class `Greetings` and write a method `hello()` that takes a `String` as input representing a name and displays an `"Hello <name>"`.
2. Create a `Test` class and save the file in the same folder.
3. From within the `Test` class try to use the method `hello()` from the `Greetings` class.
4. Play around with the modifiers of the method `hello()` and see what happens.
5. In the `Greetings` class, add a field called `msg` and use it within or outside the class, while playing around with the modifiers.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Objectives:

- See the difference between public vs private
- Use a class we have defined from another class.

NON-ACCESS MODIFIERS

- **static**

Fields, methods, and nested classes can be declared to be `static`.

When a class member is declared to be static, then it "belongs" to the entire class and not to a specific instance (object).

- **final**

Variables, methods, and classes can be declared to be `final`.

- **abstract**

Methods and classes can be declared to be `abstract`.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

STATIC

- We can define a field or a method to be `static` if we want it to be independent from one specific instance of the class.

<https://powcoder.com>

- A static method/field is associated *with the entire class*
Static fields are also called ***class variables***.

- A non-static method/field belongs to *an instance of the class*
Non-static fields are also called ***instance variables***.

STATIC VS NON-STATIC

```
String s = "hippos";  
String t = "elephants";  
boolean b = (s.length() == t.length());
```

`length()` is non-static method. Its execution depends on a specific string.

STATIC VS NON-STATIC

Assignment Project Exam Help
<https://powcoder.com>

```
double x = Math.PI;  
int y = Integer.parseInt("1");
```

Add WeChat powcoder

- **PI is a static field.** It belongs to the `Math` class.
- **parseInt () is a static method.** It belongs to the `Integer` class and does **not** depend on a specific object of type `Integer`.

STATIC VS NON-STATIC

	static	non-static
Associated with	entire class (one per class) https://powcoder.com	instance of a class (one per object/instance)
How to call (methods) from outside the class	ClassName.methodName() Add WeChat powcoder	obj.methodName()
How to reference (data) from outside the class	ClassName.varName	obj.varName

DEMO

Go back to the `Greetings` class and now play around with the static modifier. Both with the method and with the field.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

LOCAL VARIABLES VS FIELDS

How do they differ?

- Where to declare them:
 - Local variables are declared inside a method or a block
 - Fields (class and instance variables) are declared inside a class, but outside a method

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

LOCAL VARIABLES VS FIELDS

How do they differ?

- Scope:

where can they be accessed (called directly using the variable name)

- Local variables can be accessed only within the method or block in which they have been declared.
- class variables be accessed from any method or block in that class
- instance variables can be accessed from within the class or from non static methods of the class

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

LOCAL VARIABLES VS FIELDS

How do they differ?

- Access:
 - Local variables cannot have access modifiers. You can't access local variables from other classes or methods.
 - Field can have access modifiers. They can be accessed from methods within the class and from other classes if declared public.

<http://edayan.info/java/fields-vs-variables-in-java>

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

EXAMPLE - Student

- Useful Data:

- Name
- Student ID
- Grades
- Courses Taken

```
public class Student{  
    private String name;  
    private int studentID;  
    private double[] grades;  
    private String[] courses;  
}
```

- By itself, this code is a legal class definition

- Note:

- The class is `public`, it can be used by other classes
- The instance variables (non-static fields) are `private`, they can only be access from inside the class. If you try to access them from other classes you will get a compile-time error.

TRY IT!

How to use our new type `Student`:

- The snippet of code from before must go into a class called *Student.java*
- Now, create a second file *TestStudent.java* that goes inside the same folder as *Student.java*.
- Create a main method in the *TestStudent* class.
- Try to declare and initialize variables of type *Student* both from within the *Student* class and the *TestStudent* class.

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder



TO LOOK FORWARD TO

- We'll talk more about `final` in a couple of videos as well as after learning about inheritance next week.

Assignment Project Exam Help

<https://powcoder.com>

- In a week, we will also learn about abstract classes and methods.

Add WeChat powcoder

An orange paint roller with a red handle, positioned horizontally across the top of the slide. The roller is partially filled with orange paint, and there are orange paint splatters and drips around it. The background features faint, concentric circles and a horizontal line.

Coming Soon

Assignment Project Exam Help

In the next video:

- Constructors <https://powcoder.com>
- this [Add WeChat powcoder](#)