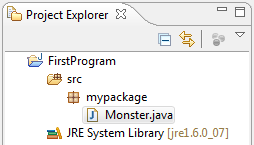
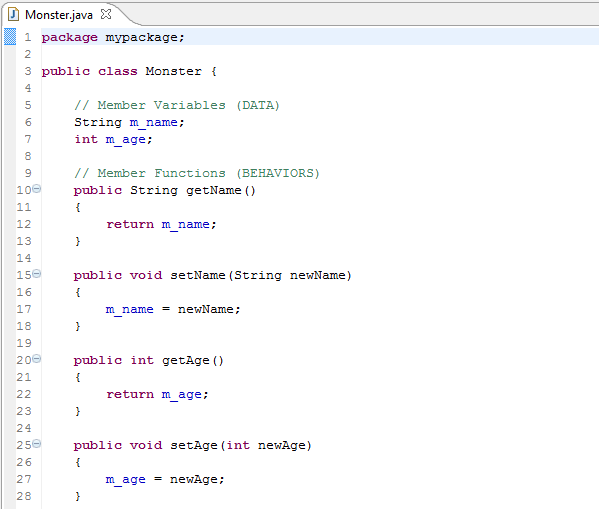
******

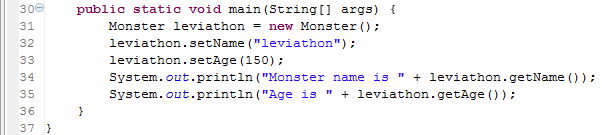
***Hands-On Exercise 1.2 [20-points]: Creating a class in Java***

***IDE structure:***



1.❑ Type the code below and get it to work.





2.❑ Paste your code here.

**package** myPackage;

**public** **class** Monster {

String m\_name;

**int** m\_age;

**public** String getM\_name() {

**return** m\_name;

}

**public** **void** setM\_name(String m\_name) {

**this**.m\_name = m\_name;

}

**public** **int** getM\_age() {

**return** m\_age;

}

**public** **void** setM\_age(**int** m\_age) {

**this**.m\_age = m\_age;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Monster leviathon = **new** Monster();

leviathon.setM\_name("Leviathon");

leviathon.setM\_age(150);

System.***out***.println("Monster name is " + leviathon.getM\_name());

System.***out***.println("Monster age is " + leviathon.getM\_age());

}

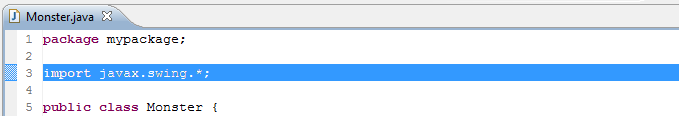
3.❑ Paste your screen shot output here [Ctrl] + [PrtScn]. Make sure you magnified it.

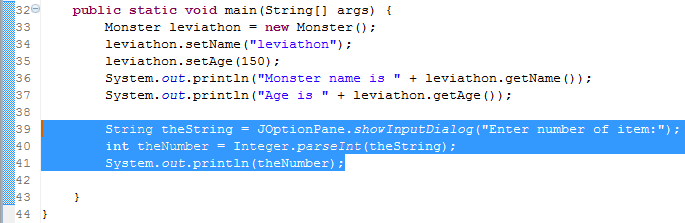
.



***Inputting data in Java:***

4.❑ Insert the lines below in the proper location.





**A note about retrieving previous Java project from within Eclipse:**

1. Open a new project, call it any name you want.
2. Import > General > File system.
3. Browse through your save project, select the folder level.
4. Check all files.

<http://itscommonsensestupid.blogspot.com/2009/01/open-existing-project-in-eclipse.html>

5.❑ Paste your code here.

**package** myPackage;

**import** javax.swing.\*;

//import javax.swing.JOptionPane;

**public** **class** Monster {

String m\_name;

**int** m\_age;

**public** String getM\_name() {

**return** m\_name;

}

**public** **void** setM\_name(String m\_name) {

**this**.m\_name = m\_name;

}

**public** **int** getM\_age() {

**return** m\_age;

}

**public** **void** setM\_age(**int** m\_age) {

**this**.m\_age = m\_age;

}

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Monster leviathon = **new** Monster();

leviathon.setM\_name("Leviathon");

leviathon.setM\_age(150);

System.***out***.println("Monster name is " + leviathon.getM\_name());

System.***out***.println("Monster age is " + leviathon.getM\_age());

String theString = JOptionPane.*showInputDialog*("Enter number of item:");

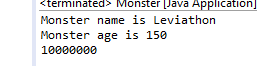
**int** theNumber = Integer.*parseInt*(theString);

System.***out***.println(theNumber);

}

}

6.❑ Paste your screen shot output here [Ctrl] + [PrtScn]. Make sure you magnified it.

. 

7.❑ Write your topmost question regarding this topic.

In .NET member variables are prefixed with “\_” such as “\_name”. Is there a similar Java convention?

8.❑ **Critical Thinking:** If you are asked to make a test question based on this topic, what would be the question and what is your answer?

What is wrong with this code?

mario.jump();

[](http://images.google.com/imgres?imgurl=www.skyscript.co.uk/im/trophy.jpg&imgrefurl=http://www.skyscript.co.uk/im/&h=214&w=180&sz=6&tbnid=ECCiP8U-7NsJ:&tbnh=99&tbnw=84&prev=/images?q=trophy&svnum=10&hl=en&lr=&ie=UTF-8&oe=UTF-8&sa=G)Congratulations! You’ve just learned how to create a class and instantiate an object based on that class in Java.

**Submission Procedure**

1. Write your **name** here: \_\_\_Joshua LeGoff\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Date: \_\_10/6/2015\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. **Backup** your work to your USB drive, this material may come out as part of your exam.
4. **Submit** to Blackboard at the link where you got it.

**Note:**

* Submit back to Blackboard where you get it.
* 2-points deduction if you submit it on the wrong place.
* 2-points deduction if you did not follow these instructions.
* Make sure you submit it at the correct location where you got it.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| GRADING RUBRIC | | | | |
| Grading Criteria | 3  **Exceeds**  *Excellent*  Epic Wow | 2  **Meets**  *Satisfactory*  O.K. | 1  **Partially Meets**  *Below Expectations*  Not Yet | 0  **Does Not Meet**  *Unacceptable*  Fail |
| **Completeness** | +5-Completed all the required work and added more examples. | +2-Completed all the work required. | +1-Partially completed the work required. | Unfortunately, did not complete the work required. |
| **Coding** | +10- Code is excellent, comments are added, and different techniques were used. | +7-Code is O.K., and program works. | +4-Code works, but still needs improvement. | Unfortunately, no coding. |
| **Output** | +5-Outputs are correct, and provided additional output cases. | +2-Output meets requirement and is readable. | +1-There is output, but not readable, and/or needs improvement. | Unfortunately, no output. |
| **Late** | Excellent, you submitted it before the deadline. | -5, unfortunately for submitting after the deadline. | -7, unfortunately for submitting several weeks after the deadline. | -10, unfortunately, for submitting very late. |