**YouTube tutorial 36 – Time class**

**1st class – apples.java**

**class** apples{

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

tuna tunaObject=**new** tuna();

System.*out*.println(tunaObject.toMilitary());

tunaObject.setTime(13,27,6);

System.*out*.println(tunaObject.toMilitary());

}

}

**2nd class – tuna.java**

**public** **class** tuna{

**private** **int** hour;

**private** **int** minute;

**private** **int** second;

**public** **void** setTime (**int** h, **int** m, **int** s){

hour=((h>=0 && h<24) ? h:0);

minute=((m>=0 && m<60) ? m:0);

second=((s>=0 && s<60) ? s:0);

}

**public** String toMilitary(){

**return** String.*format*("%02d:%02d:%02d", hour, minute, second );

}

}

“%02d” stands for 2 digits.

The “format” method is a built-in method that can only be used through String.

**YouTube tutorial 37 – Display regular time**

**1st class – apples.java**

**class** apples{

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

tuna tunaObject=**new** tuna();

System.*out*.println(tunaObject.toMilitary());

tunaObject.setTime(13,27,6);

System.*out*.println(tunaObject.toMilitary());

System.*out*.println(tunaObject.toString());

}

}

**2nd class – tuna.java**

**public** **class** tuna{

**public** **int** hour;

**public** **int** minute;

**public** **int** second;

**public** **void** setTime (**int** h, **int** m, **int** s){

hour=((h>=0 && h<24) ? h:0);

minute=((m>=0 && m<60) ? m:0);

second=((s>=0 && s<60) ? s:0);

}

**public** String toMilitary(){

**return** String.*format*("%02d:%02d:%02d", hour, minute, second );

}

**public** String toString(){

**return** String.*format*("%d:%02d:%02d %s", ((hour==0||hour==12)?12:hour%12), minute, second, (hour < 12? "AM":"PM"));

}

}

**Result:**

00:00:00

13:27:06

1:27:06 PM

**Regarding the digit code:**

%d      : will print the integer as it is.

%6d    : will print the integer as it is. If the number of digits is less than 6, the output will be padded on the left.

%-6d  : will print the integer as it is. If the number of digits is less than 6, the output will be padded on the right.

%06d : will print the integer as it is. If the number of digits is less than 6, the output will be padded on the left with zeroes.

%.2d : will print maximum 2 digits of the integer.﻿

**About the % operator in the “?12:hour%12”:**

So this is how it works. Effectively it tries to divide a number and when the number does not divide properly it wil result in its reminder. In this example let’s say 18 is been used for hour. 18 divided by 12.... so 12 fits only one time and the remainder is left as 6 (18 - 12 = 6). So with let’s say that the "Hour" has been set as 20, then we would have have 8 as remainder.

Now where I really got confused was where is if the the Hour is set anything less than 12, like why does it then return that number as a result. like if user sets Hour to 6... so it would lood like this 6/12 =??? Like why this results six? Well it’s because how many times does the 12 fit inside 6...the answer is 0.... and what is left as a reminder from 0 to 6 = the answer is 6>