

OBJECT ORIENTED PROGRAMMING



CREATED BY- VINAY A. MAHAMUNI

ESSENTIAL INGREDIENTS

- ☐ Class
- ☐ Object
- ☐ Inheritance
- ☐ Polymorphism
- ☐ Abstraction
- ☐ Encapsulation





“Any fool can write code that a
computer can understand.

Good programmers write code
that humans can understand”

–Martin Fowler



ENUM



- any time you need to represent a fixed set of constants
- Ex ["Monday", "Tuesday", ..., "Sunday"]



SWITCH IS BAD ??

```
foreach (var animal in zoo) {  
    switch (typeof(animal)) {  
        case "dog":  
            echo animal.bark();  
            break;  
  
        case "cat":  
            echo animal.meow();  
            break;  
    }  
}
```

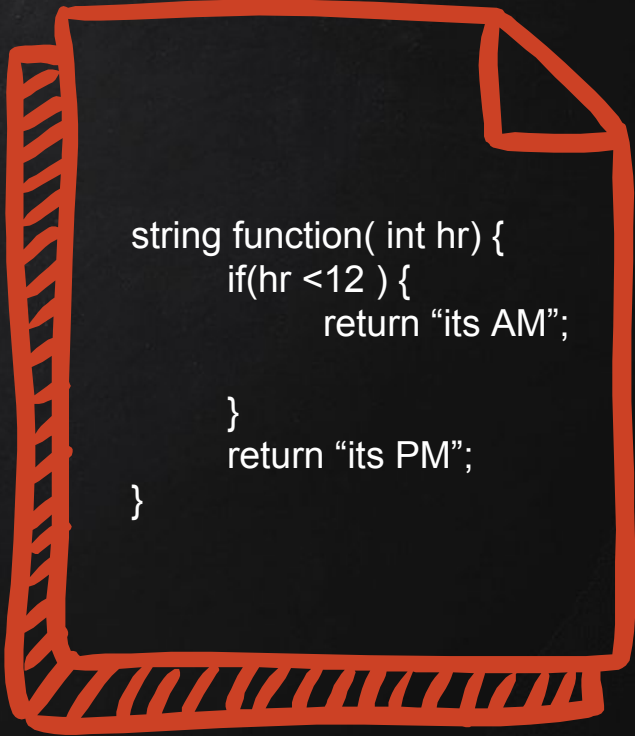


```
abstract class Animal {  
    abstract String talk();  
}  
  
class Cat extends Animal {  
    String talk() { return "Meow!"; }  
}  
  
class Dog extends Animal {  
    String talk() { return "Woof!"; }  
}  
  
int main() {  
    foreach (var animal in zoo) {  
        echo animal.speak();  
    }  
}
```

We can use Enum too.

Nested if-else

- It increase ambiguity in code
- Try to remove it.
- Can use early return



```
string function( int hr) {  
    if(hr <12 ) {  
        return "its AM";  
  
    }  
    return "its PM";  
}
```

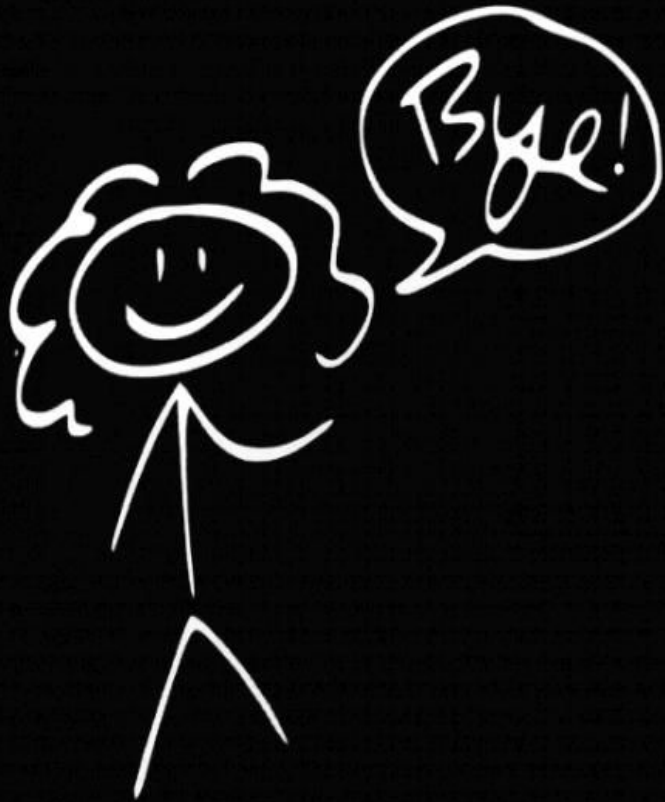
BIG **NO** TO MAGIC NUMBERS

```
for( i=0 ; i<24 ; i++)  
{  
    print(i + "th hour");  
}
```

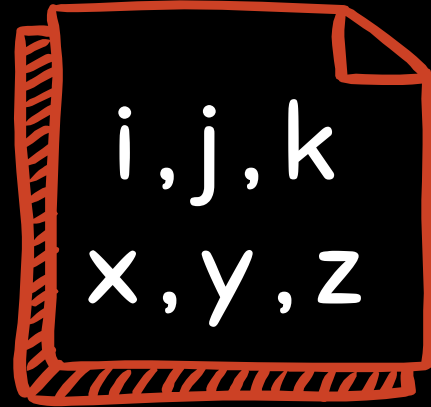


```
oneDayToHours=24;  
for(i=0 ; i < oneDayToHours ; i++)  
{  
    print(i + "th hour");  
}
```





Time to say bye bye to
your best buddies



PLEASE GIVE SOME SPACE



Proper Indentation is must !!!!!





Use IDE for coding.

NAMING CONVENTION IS MUST

Learn naming conventions of coding languages.



“CODE IS WRITTEN TO BE READ BY HUMANS, NOT COMPUTERS”

- ❑ Packages, Classes, Methods/Functions and Variables should be named in business language.
- ❑ Just by reading, any person should understand what is use of particular class or what particular method will do.
- ❑ Same with variable, variable name should match with its purpose.
- ❑ Ex. 'mypkg' is worst name for package.
- ❑ Ex. `get_T_In_MS()` 
- ❑ Ex. `getTheLatestWebSiteAccessTimeInMilliseconds()` 

METHOD

- single method should not perform multiple tasks.
- Long methods should be break down into small methods

ONE METHOD



Single
Responsibility

DON'T REPEAT YOURSELF

- ❑ Two methods shouldn't perform same task
- ❑ No duplication of code.

```
void log(String x)
{
    System.out.println(x);
}
```

```
void debug(String x)
{
    System.out.println(x);
}
```


COMMENTS

- ❑ It's good practice to write well documented code.
- ❑ But Everything shouldn't be mentioned in comments.
- ❑ Names given for classes and methods should be self explanatory.
- ❑ Excessive comments in code is considered as **BAD PRACTICE**
- ❑ Comment is deodrant.



TESTING

- Arrange, Act, Assert
- Ex. PyUnit for Python
- Ex. JUnit for JAVA
- @test → annotation for JUnit testing library

LAW OF DEMETER

Consider two class A and B and we have object of those classes a & b respectively. Then, `a.b.method()` is bad coding practice.

`a.Method()` in which internally `b.method()` is getting called is good.

Simply, use only one dot `.` for calling any function.

DESIGN PATTERN

- Command Pattern
- Factory Pattern
- Observer pattern
- Strategy pattern
- Builder pattern
- Decorator pattern
- Bridge pattern
- Visitor pattern

- tried and tested way of organizing object interactions
- code become more predictable and readable



ITS GUIDELINE
NOT RULES



LINKS FOR REFERENCE

- <http://binstock.blogspot.in/2008/04/perfecting-oos-small-classes-and-short.html>
- <http://www.davesquared.net/2009/01/introduction-to-solid-principles-of-oo.html>
- <http://priyaaank.tumblr.com/post/95095165285/decoding-thoughtworks-coding-problems>
- <https://sourcemaking.com/>



THANKS!

Any questions?



Email : vinaymahamuni22@gmail.com

Contact no : +91 77700 90795

Github link : <https://git.io/vSm2g>

