**OOPS ASSIGNMENT**

**Q1. What is inheritance in java ?  
Ans.** The process of acquiring properties of one class to another class by using of extend keyword is called as inheritance in java.

**Q2. What is superclass and subclass?  
Ans. Superclass :-**  The data acquire from the class which is known as superclass .It is also known as parent class , base class too.  
 **Subclass** :- The data acquire into inside the class which is known as subclass. It is also known as child class ,derived class too.  
 example:-   
 **class B extends A**  
class A will be parent class.  
class B will be child class

**Q3. How is inheritance implemented/achieved in java?  
Ans.** By the use of extend keyword , which is use for developing the relationship between two classes and interfaces.

**Q4. What is polymorphism?  
Ans.** If anything exists from one form to another then it is called as polymorphism. Basically it’s a Greek word in which **poly** means – many and **morphisms** mean – structure , form.

**Q5. Differentiate between method overloading and overriding?  
Ans.** The difference between overloading and overriding are:-  
**Overriding:-**

* It is like dynamic polymorphism.
* Implements on runtime.
* Error will show at runtime.
* The method name and signature should be same.
* It is derived or occur between superclass and subclass.

**Overloading:-**

* It is like static polymorphism.
* Implements on compile time.
* Error will show at compile time.
* The method name should be same but having different parameters.
* The method would be in the class.

**Q6. What is an abstraction explained with an example?  
Ans.** The method which has its only signature not a body is known as abstraction or we can say whenever in any code only the method is being used but its body would not, then it is called as abstract. If there is any method is abstract then we have to make our class also abstract.  
**Example:-**

Abstract class Animal

{  
 abstract void sleep();

{  
 // system.out.println(“Animal eats daily”);

}   
 **this body will be abstract because in child class it is override.**   
}

**Q7. What is the difference between an abstract method and final method in java. Explain with an example?  
Ans.** The abstract method is called as incomplete method because method has its own signature only but no body where as final method is complete they have signature as well as its body too.  
Once the final method is inherit it will not be override again where as we can do override the abstract method.

**Q8. What is the final class in java?  
Ans.** If the class is marked as final then it will won’t participate in inheritance. If we will try to inherit then it will show as compile time error.  
ex:- String , StringBuffer , StringBuilder , Variable, etc.

**Q9. Differentiate between abstraction and encapsulation?  
Ans.** **Abstraction:-**

* It is a feature of oops which hide the necessary info but show the essential info.
* It solves an issue at design level.
* It focused in the external lookout.
* It can be implemented by using abstract classes and interface.
* It is the process of gaining info.
* In abstraction , we use abstract classes and interfaces to hide the code complexities.
* The objects are encapsulated that helps to perform abstraction.

**Encapsulation:-**

* It is a feature of oops which hide the code and data into a single entity or unit so that the code will hide from the outside world.
* It solves an issue at implementation level.
* It focused in the internal lookout.
* It can be implemented by using access modifiers (public , protected , private).
* It is the process of containing info.
* In abstraction , we use setters and getters method to hide the data.
* The objects need not to abstract that result in encapsulation.

**Q10. Differentiate between runtime and compile time polymorphism. Explain with an example?  
Ans. Compile time polymorphism:-**

* Static polymorphism is also known as compile time polymorphism.
* Method overloading will be a type of compile time polymorphism.
* In compile time implemented should be at compile time
* Error would be come at compile time.
* The signature can be same but the parameter should not.
* With or without inheritance we can perform method overloading.

**Run time polymorphisms:-**

* It is also known as dynamic polymorphism.
* Method overriding is a type of runtime polymorphism.
* Implements on runtime.
* Error will show at runtime.
* The method name and signature should be same.
* It is derived or occur between superclass and subclass.
* With inheritance we can do only overriding.