NAME_____

29 Multiple choice questions

1. What is Method Overloading?

- In method overloading, which overloaded method is to be called is decided at compile-time itself. As decided at compile-time (and not at runtime), the performance increases because at runtime, simply methods are called (as already decided at compile-time). This is known as "static binding".
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- Declaring the same method with different parameters in the same class is known as "method overloading", a concept supported by C++ and Java but not by C-lang. Compiler differentiates which overloaded method is to be called depending on the parameter list - their number and sequence of parameters. Method overloading leads to "static polymorphism"
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- The static binding leads to static polymorphism. The same method called at different times gives different outputs - the same calculateArea() method my give the area of different geometrical figures like square and circle etc. This is known as poly (many) morphism (forms). As polymorphism is achieved with static binding, this known as static polymorphism and the other is known as "dynamic polymorphism" which Java supports.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- If the super class and the subclass have the same method, we say the super class method is overridden by subclass. This is known as "method overriding".
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

2. What are the advantages of a constructor?

- 1. Constructor is called implicitly when an object is created. Regarding method, it should be called explicitly. 2. Constructor gives properties to an object at the time of creation itself (if not, methods should be used to set the properties). 3. Programmer uses constructor to initialize variables, creating objects, setting background and foreground colors etc.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- These two are used with respect to constructors only. "this()" is used to access one constructor from another within the same class. "super()" is used to access super class constructor from subclass constructor.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A variable declared inside a method is known as "local variable" and its usage is limited within the method in which it is declared only. A "global variable" (known as instance variable in Java) is declared outside of any method but wihtin the class. Its scope is for the entire class; that is, any method can use it.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A constructor looks like a method but without return type. Moreover, the name of the constructor and class name should be the same.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

3. What is Method?

- A class is structure which holds the code and delimits (separates) the code from other classes. A class demands the programmer to write any code within the braces of the class (not outside the braces) and for this reason the main() is put within the class.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- In method overloading, which overloaded method is to be called is decided at compile-time itself. As decided at compile-time (and not at runtime), the performance increases because at runtime, simply methods are called (as already decided at compile-time). This is known as "static binding".
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- "An instance of a class is known as an object" like an instance (object) of wood is a door or table. Without object (like door or table), the wood is mere waste and similarly without object the class is waste. Remember, with object only we access a constructor, call a method etc.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A method (known familiarly as function is C-lang) is a container that holds a block of statements delimited by braces. Java categorizes the methods into static, non-static (known as concrete), final, abstract and synchronized depending on the coding needs.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

4. What is a final method?

- A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A final method in the super class cannot be overridden by the subclass. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A final class cannot be extended or inherited by any other class. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A constructor looks like a method but without return type. Moreover, the name of the constructor and class name should be the same.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

5. What is super keyword?

- Inheritance is an OOPS concept where an object one class is permitted to call other class methods. This will not come free of cost. These two classes should be linked together. The linking keyword is "extends" incase of concrete and abstract clases or "implements" incase of interfaces. When linked, the two classes establih a relation known as "inheritance". Inheritance increases code reusability (the other way is composition).
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- "final" is a keyword used as an access modifier (other popular modifiers are static and abstract; remember, access specifiers and access modifiers are different). "final" can be applied to variables, methods and classes and in each place it works (behaves) very differently.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- In method overloading, which overloaded method is to be called is decided at compile-time itself. As decided at compile-time (and not at runtime), the performance increases because at runtime, simply methods are called (as already decided at compile-time). This is known as "static binding".
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- In case of method overriding, the subclass object prefers to call its own method. That is, as along as subclass class has the same method, the super class method cannot be called by the subclass object. Otherway, we can say, the superclass method is hidden by subclass method. If the subclass also wants the superclass method, it can use "super" keyword. super keyword can be applied to variables and methods but not to classes.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

6. What is an abstract method?

- A final method in the super class cannot be overridden by the subclass. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A class that cannot be instantiated permitting the presence of concrete and abstract methods.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A method that does not contain a body.

- In method overloading, which overloaded method is to be called is decided at compile-time itself. As decided at compile-time (and not at runtime), the performance increases because at

runtime, simply methods are called (as already decided at compile-time). This is known as "static binding".
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

7. What is extends and implements?

- Just like a method can be overloaded, a constructor also can be overloaded. Depending on the parameter list, appropriate overloaded constructor is created.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- If the super class and the subclass have the same method, we say the super class method is overridden by subclass. This is known as "method overriding".
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- Both are keywords used with inheritance. "extends" is used with concrete and abstract classes and "implements" is used with interfaces.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- "super" keyword cannot be used from static context. That is, it cannot be used from static methods like main().
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

8. What is Constructor overloading?

- Just like a method can be overloaded, a constructor also can be overloaded. Depending on the parameter list, appropriate overloaded constructor is created.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- If the super class and the subclass have the same method, we say the super class method is overridden by subclass. This is known as "method overriding".
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- Both are keywords used with inheritance. "extends" is used with concrete and abstract classes and "implements" is used with interfaces.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

9. What is static binding?

A method (known familiarly as function is C-lang) is a container that holds a block of statements delimited by braces. Java categorizes the methods into static, non-static (known as concrete), final, abstract and synchronized depending on the coding needs.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

In case of method overriding, the subclass object prefers to call its own method. That is, as along as subclass class has the same method, the super class method cannot be called by the subclass object. Otherway, we can say, the superclass method is hidden by subclass method. If the subclass also wants the superclass method, it can use "super" keyword. super keyword can be applied to variables and methods but not to classes.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The static binding leads to static polymorphism. The same method called at different times gives different outputs - the same calculateArea() method my give the area of different geometrical figures like square and circle etc. This is known as poly (many) morphism (forms). As polymorphism is achieved with static binding, this known as static polymorphism and the other is known as "dynamic polymorphism" which Java supports.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

In method overloading, which overloaded method is to be called is decided at compile-time itself. As decided at compile-time (and not at runtime), the performance increases because at runtime, simply methods are called (as already decided at compile-time). This is known as "static binding".
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

10. What is the restriction of using super keyword?

"super" keyword cannot be used from static context. That is, it cannot be used from static methods like main().
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

Both are keywords used with inheritance. "extends" is used with concrete and abstract classes and "implements" is used with interfaces.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

These two are used with respect to constructors only. "this()" is used to access one constructor from another within the same class. "super()" is used to access super class constructor from subclass constructor.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

Just like a method can be overloaded, a constructor also can be overloaded. Depending on the parameter list, appropriate overloaded constructor is created.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

11. What is Composition?

- Creating an object of another class in our class and calling other class methods is known as "Composition". Composition increases reusability (and the other style is "inheritance"). Composition establishes "has-a" relationship.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- Whenever an instance (global) variable is called with an object, they are binded together so that the variable is accessible to that object only. Now, one location is created in the memory and the location is private to that object only; only that object can access or modfiy the value.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- "final" is a keyword used as an access modifier (other popular modifiers are static and abstract; remember, access specifiers and access modifiers are different). "final" can be applied to variables, methods and classes and in each place it works (behaves) very differently.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A method (known familiarly as function is C-lang) is a container that holds a block of statements delimited by braces. Java categorizes the methods into static, non-static (known as concrete), final, abstract and synchronized depending on the coding needs.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

12. What is Object?

- Inheritance is an OOPS concept where an object one class is permitted to call other class methods. This will not come free of cost. These two classes should be linked together. The linking keyword is "extends" incase of concrete and abstract clases or "implements" incase of interfaces. When linked, the two classes establih a relation known as "inheritance". Inheritance increases code reusability (the other way is composition).
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A method (known familiarly as function is C-lang) is a container that holds a block of statements delimited by braces. Java categorizes the methods into static, non-static (known as concrete), final, abstract and synchronized depending on the coding needs.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- An interface is a special type of abstract class where all methods should be abstract. The special feature of interface is it supports multiple inheritance.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- "An instance of a class is known as an object" like an instance (object) of wood is a door or table. Without object (like door or table), the wood is mere waste and similarly without object the class is waste. Remember, with object only we access a constructor, call a method etc.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

13. What is Data hiding?

- Inheritance is an OOPS concept where an object one class is permitted to call other class methods. This will not come free of cost. These two classes should be linked together. The linking keyword is "extends" incase of concrete and abstract clases or "implements" incase of interfaces. When linked, the two classes establih a relation known as "inheritance". Inheritance increases code reusability (the other way is composition).
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- The static binding leads to static polymorphism. The same method called at different times gives different outputs - the same calculateArea() method my give the area of different geometrical figures like square and circle etc. This is known as poly (many) morphism (forms). As polymorphism is achieved with static binding, this known as static polymorphism and the other is known as "dynamic polymorphism" which Java supports.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- The result of data binding is data hiding. When two objects are calling the same instance variable, two locations are created in the memory. Eventhough, the two objects belong to the same class and the instance variable is only one, they are not known to each other. One object cannot access other object's variable. They are hidden from each other. This is known as data hiding. It is possible only with OOPS languages.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- Whenever an instance (global) variable is called with an object, they are binded together so that the variable is accessible to that object only. Now, one location is created in the memory and the location is private to that object only; only that object can access or modfiy the value.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

14. What is an interface?

- A final class cannot be extended or inherited by any other class. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- An interface is a special type of abstract class where all methods should be abstract. The special feature of interface is it supports multiple inheritance.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A constructor looks like a method but without return type. Moreover, the name of the constructor and class name should be the same.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

15. What is an abstract class?

- A constructor looks like a method but without return type. Moreover, the name of the constructor and class name should be the same.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final class cannot be extended or inherited by any other class. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A method that does not contain a body.
- A class that cannot be instantiated permitting the presence of concrete and abstract methods. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

16. What are local and global variables?

- The inheritance relation results in two new names to classes involved. The class which extends is known as "subclass" and the other which is getting extended is known as "super class". By virtue of of inheritance, a subclass object can call super call properties (like variables, methods and constructors) but superclass cannot call subclass properites. This is one-way inheritance (one-way traffic).
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- 1. Constructor is called implicitly when an object is created. Regarding method, it should be called explicitly. 2. Constructor gives properties to an object at the time of creation itself (if not, methods should be used to set the properties). 3. Programmer uses constructor to initialize variables, creating objects, setting background and foreground colors etc.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A variable declared inside a method is known as "local variable" and its usage is limited within the method in which it is declared only. A "global variable" (known as instance variable in Java) is declared outside of any method but wihtin the class. Its scope is for the entire class; that is, any method can use it.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- Declaring the same method with different parameters in the same class is known as "method overloading", a concept supported by C++ and Java but not by C-lang. Compiler differentiates which overloaded method is to be called depending on the parameter list - their number and sequence of parameters. Method overloading leads to "static polymorphism"
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

17. What is a Constructor?

- A final method in the super class cannot be overridden by the subclass. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A constructor looks like a method but without return type. Moreover, the name of the constructor and class name should be the same. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final class cannot be extended or inherited by any other class. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

18. What is a class?

- A method (known familiarly as function is C-lang) is a container that holds a block of statements delimited by braces. Java categorizes the methods into static, non-static (known as concrete), final, abstract and synchronized depending on the coding needs. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A class is structure which holds the code and delimits (separates) the code from other classes. A class demands the programmer to write any code within the braces of the class (not outside the braces) and for this reason the main() is put within the class. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- "final" is a keyword used as an access modifier (other popular modifiers are static and abstract; remember, access specifiers and access modifiers are different). "final" can be applied to variables, methods and classes and in each place it works (behaves) very differently. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A constructor looks like a method but without return type. Moreover, the name of the constructor and class name should be the same. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

19. What is a final variable?

- A final class cannot be extended or inherited by any other class. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- An interface is a special type of abstract class where all methods should be abstract. The special feature of interface is it supports multiple inheritance.

- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A final method in the super class cannot be overridden by the subclass. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

20. What is subclass and super class?

- Inheritance is an OOPS concept where an object one class is permitted to call other class methods. This will not come free of cost. These two classes should be linked together. The linking keyword is "extends" incase of concrete and abstract clases or "implements" incase of interfaces. When linked, the two classes establih a relation known as "inheritance". Inheritance increases code reusability (the other way is composition).
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- The inheritance relation results in two new names to classes involved. The class which extends is known as "subclass" and the other which is getting extended is known as "super class". By virtue of of inheritance, a subclass object can call super call properties (like variables, methods and constructors) but superclass cannot call subclass properites. This is one-way inheritance (one-way traffic).
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- The static binding leads to static polymorphism. The same method called at different times gives different outputs - the same calculateArea() method my give the area of different geometrical figures like square and circle etc. This is known as poly (many) morphism (forms). As polymorphism is achieved with static binding, this known as static polymorphism and the other is known as "dynamic polymorphism" which Java supports.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- The result of data binding is data hiding. When two objects are calling the same instance variable, two locations are created in the memory. Eventhough, the two objects belong to the same class and the instance variable is only one, they are not known to each other. One object cannot access other object's variable. They are hidden from each other. This is known as data hiding. It is possible only with OOPS languages.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

21. What is a final class?

- A final class cannot be extended or inherited by any other class. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

- A constructor looks like a method but without return type. Moreover, the name of the constructor and class name should be the same.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final method in the super class cannot be overridden by the subclass. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

22. What is method overriding?

- If the super class and the subclass have the same method, we say the super class method is overridden by subclass. This is known as "method overriding".
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- Declaring the same method with different parameters in the same class is known as "method overloading", a concept supported by C++ and Java but not by C-lang. Compiler differentiates which overloaded method is to be called depending on the parameter list - their number and sequence of parameters. Method overloading leads to "static polymorphism"
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- Both are keywords used with inheritance. "extends" is used with concrete and abstract classes and "implements" is used with interfaces.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

23. What is static polymorphism?

- The data binding leads to data hiding and data hiding leads to encasulation. The data hiding can be compared to encasulation. Encapsulation is a pharmaceutical term where a drug is placed inside a gelatine capsule. By putting the drug inside the capsule, the Pharmacist hides the properties of the drug like its color and taste. Similarly in data hiding, the value associated with one object is hidden from other objects. This is known as "encapsulation" where the data is encapsulated with object, one of the OOPS concepts and the other two are inheritance and polymorphism. One final word: "the result of encapsulation is one instance (global) variable can be given multiple values (of course, by calling with different objects) which is not possible with procedural languages like C".
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The static binding leads to static polymorphism. The same method called at different times gives different outputs - the same calculateArea() method my give the area of different geometrical figures like square and circle etc. This is known as poly (many) morphism (forms). As polymorphism is achieved with static binding, this known as static polymorphism and the other is known as "dynamic polymorphism" which Java supports.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The result of data binding is data hiding. When two objects are calling the same instance variable, two locations are created in the memory. Eventhough, the two objects belong to the same class and the instance variable is only one, they are not known to each other. One object cannot access other object's variable. They are hidden from each other. This is known as data hiding. It is possible only with OOPS languages.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The inheritance relation results in two new names to classes involved. The class which extends is known as "subclass" and the other which is getting extended is known as "super class". By virtue of of inheritance, a subclass object can call super call properties (like variables, methods and constructors) but superclass cannot call subclass properites. This is one-way inheritance (one-way traffic).
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

24. What is Data binding?

Whenever an instance (global) variable is called with an object, they are binded together so that the variable is accessible to that object only. Now, one location is created in the memory and the location is private to that object only; only that object can access or modfiy the value.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

Creating an object of another class in our class and calling other class methods is known as "Composition". Composition increases reusability (and the other style is "inheritance"). Composition establishes "has-a" relationship.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The result of data binding is data hiding. When two objects are calling the same instance variable, two locations are created in the memory. Eventhough, the two objects belong to the same class and the instance variable is only one, they are not known to each other. One object cannot access other object's variable. They are hidden from each other. This is known as data hiding. It is possible only with OOPS languages.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

"final" is a keyword used as an access modifier (other popular modifiers are static and abstract; remember, access specifiers and access modifiers are different). "final" can be applied to variables, methods and classes and in each place it works (behaves) very differently.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

25. Does Java support multiple inheritance?

- Basically, Java does not support multiple inheritance but supports partially through interfaces.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A class that cannot be instantiated permitting the presence of concrete and abstract methods. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- A final method in the super class cannot be overridden by the subclass. - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

26. What are the restrictions of using this() and super()?

- A variable declared inside a method is known as "local variable" and its usage is limited within the method in which it is declared only. A "global variable" (known as instance variable in Java) is declared outside of any method but wihtin the class. Its scope is for the entire class; that is, any method can use it.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- These two are used with respect to constructors only. "this()" is used to access one constructor from another within the same class. "super()" is used to access super class constructor from subclass constructor.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- 1. Constructor is called implicitly when an object is created. Regarding method, it should be called explicitly. 2. Constructor gives properties to an object at the time of creation itself (if not, methods should be used to set the properties). 3. Programmer uses constructor to initialize variables, creating objects, setting background and foreground colors etc.
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf
- "super" keyword cannot be used from static context. That is, it cannot be used from static methods like main().
  - See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

27. What is final keyword?

- Whenever an instance (global) variable is called with an object, they are binded together so that the variable is accessible to that object only. Now, one location is created in the memory

and the location is private to that object only; only that object can access or modfiy the value.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

A final variable works like a constant of C-lang. Java does not support "const" keyword and in its place it uses final. A final variable cannot be reassigned.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

In case of method overriding, the subclass object prefers to call its own method. That is, as along as subclass class has the same method, the super class method cannot be called by the subclass object. Otherway, we can say, the superclass method is hidden by subclass method. If the subclass also wants the superclass method, it can use "super" keyword. super keyword can be applied to variables and methods but not to classes.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

"final" is a keyword used as an access modifier (other popular modifiers are static and abstract; remember, access specifiers and access modifiers are different). "final" can be applied to variables, methods and classes and in each place it works (behaves) very differently.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

28. What is inheritance?

Inheritance is an OOPS concept where an object one class is permitted to call other class methods. This will not come free of cost. These two classes should be linked together. The linking keyword is "extends" incase of concrete and abstract clases or "implements" incase of interfaces. When linked, the two classes establih a relation known as "inheritance". Inheritance increases code reusability (the other way is composition).
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The inheritance relation results in two new names to classes involved. The class which extends is known as "subclass" and the other which is getting extended is known as "super class". By virtue of of inheritance, a subclass object can call super call properties (like variables, methods and constructors) but superclass cannot call subclass properites. This is one-way inheritance (one-way traffic).
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

In case of method overriding, the subclass object prefers to call its own method. That is, as along as subclass class has the same method, the super class method cannot be called by the subclass object. Otherway, we can say, the superclass method is hidden by subclass method. If the subclass also wants the superclass method, it can use "super" keyword. super keyword can be applied to variables and methods but not to classes.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The result of data binding is data hiding. When two objects are calling the same instance

variable, two locations are created in the memory. Eventhough, the two objects belong to the same class and the instance variable is only one, they are not known to each other. One object cannot access other object's variable. They are hidden from each other. This is known as data hiding. It is possible only with OOPS languages.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

29. What is Encapsulation?

The result of data binding is data hiding. When two objects are calling the same instance variable, two locations are created in the memory. Eventhough, the two objects belong to the same class and the instance variable is only one, they are not known to each other. One object cannot access other object's variable. They are hidden from each other. This is known as data hiding. It is possible only with OOPS languages.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The data binding leads to data hiding and data hiding leads to encasulation. The data hiding can be compared to encasulation. Encapsulation is a pharmaceutical term where a drug is placed inside a gelatine capsule. By putting the drug inside the capsule, the Pharmacist hides the properties of the drug like its color and taste. Similarly in data hiding, the value associated with one object is hidden from other objects. This is known as "encapsulation" where the data is encapsulated with object, one of the OOPS concepts and the other two are inheritance and polymorphism. One final word: "the result of encapsulation is one instance (global) variable can be given multiple values (of course, by calling with different objects) which is not possible with procedural languages like C".
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

The static binding leads to static polymorphism. The same method called at different times gives different outputs - the same calculateArea() method my give the area of different geometrical figures like square and circle etc. This is known as poly (many) morphism (forms). As polymorphism is achieved with static binding, this known as static polymorphism and the other is known as "dynamic polymorphism" which Java supports.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

Whenever an instance (global) variable is called with an object, they are binded together so that the variable is accessible to that object only. Now, one location is created in the memory and the location is private to that object only; only that object can access or modfiy the value.
- See more at: http://way2java.com/java-questions/oops/#sthash.N9rsu6On.dpuf

Check answers