Object Oriented Programming (with JAVA) UNIT-2 Question Bank
by Dr. Partha Roy, Asso. Prof., BIT, Durg

****

1. Explain with proper example UML Class notation.
2. Explain with proper example the concept of Concrete class.
3. Explain with proper example the concept of Abstract class.
4. Explain with proper example the concept of Interface.
5. Compare Abstract class and Interface.
6. WAP to demonstrate any four properties of Concrete classes.
7. WAP to demonstrate any four properties of Abstract classes.
8. WAP to demonstrate any four properties of Interface.
9. Compare Association, Aggregation and Composition.
10. Explain with proper example the concept of Association in Java.
11. Explain with proper example the concept of Aggregation in Java.
12. Explain with proper example the concept of Composition in Java.
13. Explain with proper example the concept of "extends" and "implements" in Java.
14. Explain with proper example the concept of Member access within and outside package in Java.
15. Compare Private, Default, Protected and Public access modes in Java.
16. Explain with proper example the concept of Single Level Inheritance in Java.
17. Explain with proper example the concept of Multilevel Inheritance in Java.
18. Explain with proper example the concept of Hierarchical Inheritance in Java.
19. Explain with proper example the concept of Multiple Inheritance in Java.
20. Explain with proper example how Static members behave during inheritance.
21. Explain with proper example the use of "super" reference and "super()" method in Java.
22. Explain with proper example the concept of Method overloading in Java.
23. Explain with proper example the concept of Method overriding in Java.
24. Compare Method overloading and Method overriding in Java.
25. Explain with proper example the concept of Dynamic Binding in Java.
26. Explain with proper example the concept of Singleton classes in Java.
27. Explain with proper example the concept of Exception and Error in Java.
28. Explain with proper example the use of try-catch-finally in Java.
29. Explain with proper example the use of throw and throws in Java.
30. Explain with proper example the concept of Checked Exception (both inbuilt and user defined) in Java.
31. Explain with proper example the concept of Un-Checked Exception (both inbuilt and user defined) in Java.
32. Explain with proper diagram the Exception hierarchy of classes in Java.
33. Explain with proper flowchart the concept of Exception handling process by JVM.
34. Write a program in Java to demonstrate inheritance within a package and another program to demonstrate inheritance across packages.
35. Write a program in Java to demonstrate method overloading and method overriding.
36. Write a program in Java to show how the constructors are called in Multilevel inheritance. The parent classes should contain only parameterized constructors (no default constructors).
37. Write a program in Java to show how the constructors are called in Hierarchical inheritance. The parent class should contain only parameterized constructors (no default constructors).
38. Write a program in Java to demonstrate the inheritance of final data members and member methods during Multilevel inheritance.
39. Write a program in Java to demonstrate the inheritance of static data members and member methods during Multilevel inheritance.

40. Complete the program. Class A is the parent of class B and class B is the parent of class C. All the classes should only contain parameterized constructors.

```
public class Test{
        public static void main(String arg[]){
                C ob=new C(10,20,30);
                //30 should reach class A,20 should reach class B and 10 should reach class C
        }
}
```

Output:
Class-A=30
Class-B=20
Class-C=10

41. Complete the program. Class A is the parent of class B and class C. All the classes should only contain parameterized constructors.

```
public class Test{
        public static void main(String arg[]){
                B ob=new B(10,20);//10 should reach class A and 20 should reach class B
                C ob=new C(30,40);//30 should reach class A and 40 should reach class C
        }
}
```

Output:
Class-A=10
Class-B=20
Class-A=30
Class-C=40

42. Write a program in Java to create a Shape Interface which has a member method area(). Derive two subclasses Circle and Triangle from it. Using reference of Shape class fill the required members in Circle and Triangle also display the area of Circle and Triangle. Take input from user while filling data members.
43. Write a program in Java to demonstrate the two ways in which singleton classes can be created and use factory methods from those classes to create instance of the singleton classes.
44. Class A is a singleton class. Complete the following program:

```
public class NewClass {
   public static void main(String arg[]) {
      A ob1,ob2,ob3; // objects can be declared but new instances cannot be created
      ob1=A.getA(); //getA() is a factory method
      ob2=A.getA();
      ob3=A.getA();
      ob1.setN(100);
      System.out.println(ob1.getN()+" "+ob2.getN()+" "+ob3.getN());
      ob2.setN(200);
      System.out.println(ob1.getN()+" "+ob2.getN()+" "+ob3.getN());
      ob3.setN(300);
      System.out.println(ob1.getN()+" "+ob2.getN()+" "+ob3.getN());
   }
}
```

Output:
100 100 100
200 200 200
300 300 300

45. Write a program in Java to generate a customized un-checked exception whenever a number input from the user is negative.
46. Write a program in Java to generate a customized checked exception whenever a number input from the user is either less than 20 or greater than 50.