Core Java Day-3

**Q.** Which of the following statements are true related to interface ?

**1)**. **Interface doesn't alllow to create object .**

**2)**. **Multiple inheritence can be possible in interface.**

**3)**. Class can implement only one interface .

**4)**. Interface can't allow to declare a member variable.

**Q.** On which of the given options abstract modifier can be used?

**1)**. constructor

**2)**. static method

**3)**. **non-static methods**

**4)**. **class**

**Q.** Consider the following code:   
  
interface Greek { }   
  
class Alpha implements Greek { }   
  
class Beta extends Alpha {}   
  
class Delta extends Beta   
 {   
    public static void main( String[] args )   
     {   
       Beta obj = new Beta(); // insert code here   
     }  
 }  
  
Which of the following code snippet when inserted individual at the commented line   
(// insert code here), will cause a  java.lang.ClassCastException?

**1)**. Greek objGrk = (Beta)(Alpha)obj;

**2)**. Alpha objAlpha = obj;

**3)**. Greek objGrk = (Alpha)obj;

**4)**.  Beta objBeta = (Beta)(Alpha)obj;

**5)**. **Greek objGrk = (Delta)obj;**

**Q.** Consider the following code:   
  
interface MyInterface {  
  
// Method declaration code  
  
 }   
Which of the following code snippet shows the wrong way to declare Method in interface ?

**1)**. public abstract boolean isValid();

**2)**. public boolean isValid();

**3)**. **protected boolean isValid();**

**4)**.  boolean isValid();

**Q.** public class Demos {  
        
      public static void main(String[] args) {  
       String input = "Hello Welcome";  
       String pattern = "\\sHello\\sWelcome\\s";  
       boolean flag = Pattern.matches(pattern, input);  
       System.out.println(flag);  
      }  
}

**1)**. TRUE

**2)**. **FALSE**

**3)**. compilation error

**4)**. exception

**Q.** public class RegexMatches  
{  
    private static String regex = "dog";  
    private static String input = "The dog says meow. " +"All dogs say meow.";  
    private static String replace = "cat";  
  
    public static void main(String[] args) {  
        //line no.1  
         
       Matcher m = p.matcher(input);   
       INPUT = m.replaceAll(replace);  
       System.out.println(input);  
   }  
}Which code need to be inserted at line no.1 to execute it successfully?

**1)**. **Pattern p = Pattern.compile(regex);**

**2)**. Pattern p = Pattern.test(regex);

**3)**. Pattern p = Pattern.matches(regex);

**4)**. Pattern p = new Pattern(regex);

**Q.** Which of the given syntax is correct for interface implementation in Java 8?

**1)**. **interface Demo{  
}**

**2)**. **interface Demo{  
default int getNumber(){return 0;}  
}**

**3)**. **interface Demo{  
static int getNumber(){return 0;}  
}**

**4)**. interface Demo  
{  
int getNumber(){return 0;}  
}

**Q.** Which of the following is FALSE about abstract classes in Java?

**1)**. If we derive an abstract class and do not implement all the abstract methods, then the derived class should also be marked as abstract using 'abstract' keyword

**2)**. Abstract classes can have constructors

**3)**. A class can be made abstract without any abstract method

**4)**. **A class can inherit from multiple abstract classes**

**Q.** class Shape  
{  
Shape(){}  
}  
public class Triangle extends Shape  
{  
int base,height;  
public Triangle(int base,int height){  
this.base = base;  
this.height=height;  
}  
public double calArea(){  
return o.5\*base\*height;  
}  
public static void main(String []args){  
Shape ref = new Triangle(3,4);  
System.out.println(ref.calArea());  
}Which change need to be done in the Shape class for successful execution of program?

**1)**. Declare Shape as abstract class

**2)**. Remove default constructor from Shape class

**3)**. declare abstract double calArea() in Shape class

**4)**. **declare abstract double calArea() in Shape class and declare Shape class as abstract class**

**Q.** What is the Regular Expression Matching Zero or More Specific Characters?

**1)**. \

**2)**. $

**3)**. **\***

**4)**. ^