

CORRIGE-TYPE EXAMEN

Exo 01 : (07 pt)

I-

1- . (03 pt)

```
public class Etudiant {  
    public String nom , prenom ;  
    String spicialite ;  
    public Etudiant (String nom, String prenom) {  
        this.nom = nom;  
        this.prenom = prenom ;  
    }  
    Public void setPrenom (String prenom) {this.prenom = prenom ; }  
}
```

2-

Class Prog

```
{ public static void main (String [] args )  
{  
    Etudiant e = new Etudiant (“Slimani” , “Mohamed”);    (01 pt)  
}
```

3- e.setPrenom (“Ahmed”) (01 pt)

II-

4- (01 pt)

```
Public abstract class Exo {  
    Public abstract calculer() ;  
}  
}
```

III-

5- les méthode de classe A sont : (01 pt)

```
void dessiner () {//....}  
void colorer(String couleur) {//.....}  
Public void afficherInfo() {//.....}
```

Exo 02 : (13 pt)

1- Nombre x = new Nombre(0); // pour crée un nombre x ayant la valeur 0 (01 pt)

2- à la fin de l'exécution y = 3 et z = 9 (02 pt)

3- (01 pt)

```
public void afficher () {System.out.println ("la valeur du nombre est " + value)}.
```

4- (02 pt)

```
public int getValue(){return value ;}
```

```
public void setValue (int value) { this.value = value ; }
```

5- Oui , car la classe NombreSpecial est une sous classe de Nombre Donc le upcasting est implicite (principe de polymorphisme) (01 pt)

6- (01 pt)

```
public boolean estNegatif(){
```

```
if (value < 0 )
```

```
    return true ;
```

```
else
```

```
    return false ;
```

```
/* ou bien
```

```
return (value < 0) ;
```

```
*/
```

```
}
```

7- (01 pt)

```
Public int absolu ()
```

```
{
```

```
if (value <0 )
```

```
    return - value ;
```

```
else
```

```
    return value ;
```

```
}
```

8- (01 pt)

```
public boolean estEgale (Nombre n){
```

```
if (value == n.value)
```

```
return true ;
```

```
else
```

```
    return false ;
```

```
// ou bien (value == n.value) ;
```

```
}
```

9- (01 pt)

```
public boolean estEgale (Nombre n1 , Nombre n2){
```

```
if (n1.value == n2.value)
```

```
return true ;
```

```
else
```

```
    return false ;
```

```
// ou bien (n1.value == n2.value) ;
```

```
}
```

10-. (02 pt)

```
public class suite
```

```
{
```

```
protected int taille ;
```

```
protected Nombre [] tab ;
```

```
public Suite (int taille)
```

```
{
```

```
    this.taille = taille ;
```

```
    tab = new Nombre [taille] ;
```

```
}
```

```
}
```