

```

/**
  Wat is de output van het volgend programma?
 */
public class ObjectOefening {

    /** PROGRAMMA */
    public static void main(String[] args) {

        KlasseA a1 = new KlasseA(10);
        KlasseA a2 = new KlasseA(2, 8);

        System.out.println("a1 = "+a1);
        System.out.println("a2 = "+a2);

        KlasseB b1 = new KlasseB();
        System.out.println("b1 = "+b1);

        a1.f(3);
        a2.f(3);

        System.out.println("a1 = "+a1);
        System.out.println("a2 = "+a2);

        b1.g(a2);
        System.out.println("b1 = "+b1);
    }
}

class KlasseA {
    int x, y;
    KlasseA(int x){
        this.x=x;
        this.y=0;
    }
    KlasseA(int x, int y){
        this.x=x;
        this.y=y;
    }
    void f(int factor){
        x = factor * x;
        y = y / factor;
    }
    public String toString(){
        return "A [x="+x+";y="+y+"]";
    }
}

class KlasseB {
    int z = 5; // default waarde

    void g(KlasseA a){
        z = z * a.x + a.y;
    }
    public String toString(){
        return "B [z="+z+"]";
    }
}

```

## Oplossing

<code>KlasseA a1 = new KlasseA(10);</code>	Met new wordt de constructor van KlasseA opgeroepen. Er zijn 2 constructoren (je herkent ze omdat het methodes zijn met dezelfde naam als de klasse), de eerste heeft maar 1 parameter. Dat is degene die hier wordt uitgevoerd. Deze zet attribuut x op 10 en y op 0. Variabele a1 wordt gelinkt aan dit nieuwe object.
<code>KlasseA a2 = new KlasseA(2, 8);</code>	Hier wordt de 2e constructor opgeroepen, want 2 parameters. Van dit object krijgt attribuut x de waarde 2 en y de waarde 8. Variabele a2 wordt gelinkt aan dit object. Er zijn dus 2 objecten aangemaakt van dezelfde klassen. Ze hebben andere waarden voor hun attributen.
<code>System.out.println("a1 = "+a1);</code>	Object a1 word geprint. Java zal dan automatisch de toString()-methode oproepen. Deze methode zet het object om naar een string. We krijgen dan als output: <code>"a1 = A[x=10;y=0]"</code> .
<code>System.out.println("a2 = "+a2);</code>	Nu krijgen we als output: <code>"a2 = A[x=2;y=8]"</code> .
<code>KlasseB b1 = new KlasseB();</code>	Met new wordt de constructor van KlasseB opgeroepen. Er is in de code echter geen constructor gedefinieerd. Dan zal Java automatisch de default constructor voorzien: <pre>KlasseB(){ }</pre> Die doet in feite niets. Het nieuwe object heeft 1 attribuut, namelijk z. Deze krijgt de initiele waarde 5.
<code>System.out.println("b1 = "+b1);</code>	De toString()-methode van KlasseB wordt opgeroepen. Er wordt geprint: <code>"b1 = B[z=5]"</code> .
<code>a1.f(3);</code>	De methode f wordt opgeroepen voor object a1. Die methode zal attribuut x vermenigvuldigen met de meegegeven parameter en attribuut y er door delen. X is dan 30 geworden, y blijft 0.
<code>a2.f(3);</code>	Nu gebeurt de methode op object a2 en zullen die zijn attributen veranderen. X van a2 wordt 6 en y wordt 2 ( $8/3 = 2.666$ maar omdat integers wordt alles achter de komma weggegooid).
<code>System.out.println("a1 = "+a1);</code>	<code>"a1 = A[x=30;y=0]"</code>
<code>System.out.println("a2 = "+a2);</code>	<code>"a2 = A[x=6;y=2]"</code>
<code>b1.g(a2);</code>	De methode g van b1 wordt opgeroepen en a2 wordt als parameter meegegeven. Z wordt dan berekend met de attributen x en y van a2 als volgt: $z = 5*6+2 = 32$ .
<code>System.out.println("b1 = "+b1);</code>	<code>"b1 = B[z=32]"</code>