Object

```
import java.io.*; import java.lang.*;
class point
{ private float x,y;    public point(float a,float b){x=a;y=b;}   public point(){}
  public void print() { System.out.println("("+x+","+y+")"); }
  public void abc() {x=2*x;y=3*y;}
  public static point pqr(point a)  {point t;t=new point(2*a.x,3*a.y);return t;}
  public void ghi(point a) {x=2*a.x;y=3*a.y;}  public float getx(){return x;}
}
class abc
{ private float a,b,c;   public abc(float p,float q,float r){a=p;b=q;c=r;}   public abc(){}
  public void print() { System.out.println("("+a+","+b+","+c+")"); }
}
class hari
{  public static void main(String args[])
   {   point a,b,c;float t;abc k;      a=new point(2,4);a.print();a.abc();a.print();
       a=new point(2,4);b=point.pqr(a);b.print(); a=new point(2,4);c=new point();c.ghi(a);c.print();
       t=a.getx();System.out.println(t);   k=new abc(2,4,9);k.print();
   }
}
```

In above if point p is (x,y) then p.abc( ) will make it (2x,3y).
If point r is (x,y) then q=point.pqr(r) will make q as (2x,3y). r is unmodified.
If point k is (x,y) then r.ghi(k) will make r as (2x,3y)

In the definition of following the use of local variables should be minimum.

1. Define fa. If point p is (x,y) then p.fa() will make it (x+y,2y). (20,4) → (24,8) [no local variable]
2. Define fb. If point p is (x,y) then p.fb() will make it (2x,x+y). (20,4) → (40,24) [no local variable]
3. Define fc. If point p is (x,y) then p.fc() will make it (x+y,x*y). (20,4) → (24,80) [local variable]
4. Define fd. If point p is (x,y) then q=p.fd() will make q as (x+y,x*y). [no float local variable, p unchanged]
5. Define fe. If point p is (x,y) then q=p.fe() will make q as (x+y,x*y). [no local variable]
6. Define ga. If point p is (x,y) then q=p.ga() will make q as (x+y,2y). (2,4) → (6,8)
7. Define gb. If point p is (x,y) then q=point.gb(p) will make q as (x+y,2y).
8. Define kb using ga. In its definition + and * should not be used.
   kb is similar to gb. p=(x,y) q=point.kb(p) ⇒ q=(x+y,2y)
9. Define gc. If point p is (x,y) then q.gc(p) will make q as (x+y,2y).
10. Define kc using ga. Do not use + and *. p=(x,y) q.kc(p) ⇒ q=(x+y,2y) (similar to gc)
11. Define gd. If point p is (x,y) then p.gd(k) will make it (x+k,2y). If p=(5,7) then p.gd(12) will make p as (17,14).
12. Define ha, hb and hc using gd. In their definition + and * should not be used. They are similar to ga, gb and gc. p=(x,y) q=p.ha( ) ⇒ q=(x+y,2y) q=point.hb(p) q.hc(p) Observe the output of following main program. The value of object 'a' should remain unchanged as (2,4). main( ){ a=new point(2,4); c=a.ha( ); a.print( ); c.print( ); }

In the definition of followings the use of getx, gety, geta, etc should be minimum. Care should be taken to choose the class in the definition of following functions.

13. Define faa. If point p is (x,y) then q=point.faa(p) will make q as (x+y,x*y,x−y).

14. Define fab. If point p is (x,y) then q=abc.fab(p) will make q as (x+y,x*y,x–y).
15. Define kaa using fab. Do not use +, *, –. p=(x,y) q=point.kaa(p) ⇒ q=(x+y,x*y, x–y)
16. Define kab using faa. Do not use +, * and –. p=(x,y) q=abc.kab(p) ⇒ q=(x+y,x*y, x–y)
17. Define fac. If q is (x,y,z) then p=q.fac() will make p as (x+y,y*z).
18. Define fad. If q is (x,y,z) then p.fad(q) will make p as (x+y,y*z).
19. Define kac using fad. Do not use + and *. q=(x,y,z) p=q.kac() ⇒ p=(x+y,y*z)
20. Define kad using fac. Do not use + and *. q=(x,y,z) p.kad(q) ⇒ p=(x+y,y*z)
21. In class xyz, define functions f, g and h so that following main program outputs (x,y) (x+y,2x) and (10(x+y)+1,30x+1). In function g only * and in h only + should be used. No other arithmetic operator should be used in them.
    main( ){ xyz a,b; a=new xyz(x,y); a.pt();b=xyz.h(a.f().g());a.pt();b.pt();}
22. Do above problem when in g only *5 and in h only +1 is used.
23. Do above problem without using new in function f. In function g new can be used.
24. Do above problem without using new in function g. In function f new can be used.
25. Do above problem without using new in f,g. In function f only + should be used. In function g only * should be used. In function h both * and + can be used.