/\*create 10 classes a b c d e f g h i j with at least null constructor and a parametric constructor

,3 data members and functions with different access modifiers specify the access of each data member

and function in the main program use these classes to illustrate the concept of single inheritance

multiple inheritance and hybrid inheritance\*/

#include<iostream>

using namespace std;

class A{

private:

int b=0;

protected:

int c=0;

public:

int a;

C1()

{

a = '0';

b = '0';

c = '0';

}

A(int a, int b, int c)

{

cout << "parametric Constructor is called \n";

}

};

class B{

private:

int e;

protected:

int f;

public:

int d;

B()

{

d= '0';

e = '0';

f = '0';

}

B(int d, int e, int f)

{

cout << "parametric Constructor is called of class 2 \n";

}

};

class C{

private:

int h;

protected:

int i;

public:

int g;

C()

{

g = '0';

h = '0';

i = '0';

}

C(int g, int h, int i)

{

cout << "parametric Constructor is called of class 3 \n";

}

};

class D

{

private:

int k;

protected:

int l;

public:

int j;

D()

{

k = '0';

l = '0';

j = '0';

}

D(int k, int l, int j)

{

cout << "parametric Constructor is called of class 4 \n";

}

};

class E{

private:

int n;

protected:

int o;

public:

int m;

E()

{

m = '0';

n = '0';

o = '0';

}

E(int k, int l, int j)

{

cout << "parametric Constructor is called of class 5 \n";

}

};

class F:public A{

private:

int q;

protected:

int r;

public:

int p;

F()

{

p = '0';

q = '0';

r = '0';

}

F(int p, int q, int r)

{

cout << "parametric Constructor is called of class 6 \n";

}

};

class G: public F{

private:

int t;

protected:

int u;

public:

int s;

G()

{

s = '0';

t = '0';

u = '0';

}

G(int s, int t, int u)

{

cout << "parametric Constructor is called of class 7 \n";

}

};

class H: public G{

protected:

int x;

private:

int w;

public:

int v;

H()

{

v = '0';

w = '0';

x = '0';

}

H(int v, int w, int x)

{

cout << "parametric Constructor is called of class 8 \n";

}

};

class I: public G : public H{

protected:

int a1;

private:

int z;

public:

int y;

I()

{

y = '0';

z = '0';

a1 = '0';

}

I(int y, int z, int a1)

{

cout << "parametric Constructor is called of class 9 \n";

}

};

class J: public G : public H : public I{

protected:

int a4;

private:

int a3;

public:

int a2;

J()

{

a2 = '0';

a3 = '0';

a4 = '0';

}

J(int a2, int a3, int a4)

{

cout << "parametric Constructor is called of class 10 \n";

}

};

int main(){

int main()

{

A obj1(1,2,3);

B obj2(4,5,6);

C obj3(7,8,9);

D obj4(10,11,12);

E obj5(13,14,15);

F obj6(16,17,18);

G obj7(19,20,21);

H obj8(22,23,24);

I obj9(25,26,27);

J obj10(28,29,30);

cout << obj1.a << "\n" << obj2.d << "\n" << obj3.g << "\n" << obj4.j << "\n" << obj5.m << "\n" << obj6.p << obj7.s << "\n" << obj8.v << "\n" << obj9.y << "\n" << obj10.a2 << endl;

}