**Part (a)**

public class CombinedTable

{

private SingleTable table1;

private SingleTable table2;

public CombinedTable( SingleTable st1, SingleTable st2 )

{

table1 = st1;

table2 = st2;

}

public boolean canSeat( int num )

{

int maxSeated = table1.getNumSeats() + table2.getNumSeats() - 2

if(num <= maxSeated)

return true

else

return false;

}

public double getDesirability()

{

if( table1.getHeight() == table2.getHeight() )

return (table1.getViewQuality() + table2.getViewQuality() ) / 2 ;

else

return (table1.getViewQuality() + table2.getViewQuality() ) / 2 - 10;

}

}

**//Alternative solution Part (a)**

public class CombinedTable

{

private int numCanSeat;

private double desirability:

public CombinedTable(SingleTable s1, SingleTable s2)

{

numCanSeat = s1.getNumSeats() + s2.getNumSeats() - 2;

if(s1.getHeight() == s2.getHeight())

{

desirability = s1.getViewQuality() + s2.getViewQuality() / 2;

}

else

{

desirability = (s1.getViewQuality() + s2.getViewQuality() / 2) - 10;

}

}

public boolean canSeat(int n)

{

if(n > numCanSeat)

{

return false;

}

return true;

}

public double getDesirability()

{

return desirability;

}

}