The **sports** class might look like:

```
class sports
{
  protected:
     float  score;
  public:
     void get_score(float);
     void put_score(void);
};
```

The result will have both the multilevel and multiple inheritances and its declaration would be as follows:

```
class result : public test, public sports
{
       .....
       .....
};
```

Where test itself is a derived class from student. That is

```
class test : public student
{
      .....
      .....
};
```

Program 8.5 illustrates the implementation of both multilevel and multiple inheritance.

HYBRID INHERITANCE

```
#include <iostream>

using namespace std;

class student
{
  protected:
     int  roll_number;
  public:
     void get_number(int a)
     {
          roll_number = a;
```

(Contd)

```
        }
        void put_number(void)
        {
              cout << "Roll No: " << roll_number << "\n";
        }
};

class test : public student
{
  protected:
      float part1, part2;
  public:
      void get_marks(float x, float y)
      {
            part1 = x;  part2 = y;
      }
      void put_marks(void)
      {
            cout << "Marks obtained: " << "\n"
                 << "Part1 = " << part1 << "\n"
                 << "Part2 = " << part2 << "\n";
      }
};

class sports
{
   protected:
       float score;
   public:
       void get_score(float s)
       {
            score = s;
       }
       void put_score(void)
       {
            cout << "Sports wt: " << score << "\n\n";
       }
};

class result : public test, public sports
{
       float total;
  public:
       void display(void);
```

(Contd)

```
};

void result :: display(void)
{
      total = part1 + part2 + score;

      put_number();
      put_marks();
      put_score();

      cout << "Total Score: " << total << "\n";
}

int main()
{
      result student_1;
      student_1.get_number(1234);
      student_1.get_marks(27.5, 33.0);
      student_1.get_score(6.0);
      student_1.display();

      return 0;
}
```

PROGRAM 8.5

Here is the output of Program 8.5:

```
Roll No: 1234
Marks obtained:
Part1 = 27.5
Part2 = 33
Sports wt: 6

Total Score: 66.5
```

8.9 Virtual Base Classes

We have just discussed a situation which would require the use of both the multiple and multilevel inheritance. Consider a situation where all the three kinds of inheritance, namely, multilevel, multiple and hierarchical inheritance, are involved. This is illustrated in Fig. 8.12. The 'child' has two *direct base classes* 'parent1' and 'parent2' which themselves have a common base class 'grandparent'. The 'child' inherits the traits of 'grandparent' via two separate paths. It can also inherit directly as shown by the broken line. The 'grandparent' is sometimes referred to as *indirect base class*.