```cpp
#include<iostream>
using namespace std;

class Employee {
    public:
    int empid;
    int empcode;
        virtual void displayData()
        {
            cout << "Temp";
        }
};

class Programmer : public Employee {
    public:
        string skill;
        Programmer (int id, int code, string s)
        {
            empid = id;
            empcode = code;
            skill = s;
        }
        void displayData()
        {
            cout << "EmpId: " << empid << endl;
            cout << "EmpCode: " << empcode << endl;
            cout << "Skill: " << skill << endl;
        }

};

class Manager : public Employee {
    public:
        string department;
        Manager (int id, int code, string dept)
        {
            empid = id;
            empcode = code;
            department = dept;
        }
        void displayData()
        {
            cout << "EmpId: " << empid << endl;
            cout << "EmpCode: " << empcode << endl;
            cout << "Department: " << department <<
                endl;
        }
};

int main ()
{
    Programmer p1(1, 1001, "Skill less");
    Manager m1(2, 1002, "No department");

    p1.displayData();
    m1.displayData();

    return 0;
}
```

Output:

```
/tmp/y6jzPu6UgY.o
EmpId: 1
EmpCode: 1001
Skill: Skill less
EmpId: 2
EmpCode: 1002
Department: No department


=== Code Execution Successful ===
```

```cpp
#include <iostream>
using namespace std;

class Staff {
public:
    int code;
    string name;

    void input() {...}
    void display() const {...}
};

class Teacher : public Staff {
public:
    string subject;
    string publication;

    void input() {...}
    void display() const {...}
};

class Officer : public Staff {
public:
    int grade;

    void input() {...}
    void display() const {...}
};

class Typist : public Staff {
public:
    float speed;

    void input() {...}
    void display() const {...}
};

class Regular : public Typist {
public:
    int wages2;

    void input() {...}
    void display() const {...}
};

class Casual : public Typist {
public:
    float wages;

    void input() {...}
    void display() const {...}
};

int main() {...}
```

Output

```
/tmp/pfEDWr7DDC.o
Enter details for Teacher:
Enter staff code: 20001
Enter staff name: Teacher ABC
Enter subject: Maths
Enter publication: Publication 101

Details of Teacher:
Code: 20001
Name: Teacher ABC
Subject: Maths
Publication: Publication 101

Enter details for Officer:
Enter staff code: 10001
Enter staff name: Officer ABC
Enter grade: 98

Details of Officer:
Code: 10001
Name: Officer ABC
Grade: 98

Enter details for Regular Typist:
Enter staff code: 30001
Enter staff name: Regular Typist ABC
Enter typing speed: 65
Enter regular wages: 75000

Details of Regular Typist:
Code: 30001
Name: Regular Typist ABC
Typing Speed: 65 words per minute
Regular Wages: 75000

Enter details for Casual Typist:
Enter staff code: 30102
Enter staff name: Casual Typist ABC
Enter typing speed: 110
Enter casual wages: 90000

Details of Casual Typist:
Code: 30102
Name: Casual Typist ABC
Typing Speed: 110 words per minute
Casual Wages: 90000


=== Code Execution Successful ===
```

```cpp
#include <iostream>
using namespace std;

class Player {
    public:
        string name;
        int matches;

        virtual void display()
        {
            cout << "Temp";
        }
};

class Batsman : public Player {
    public:
        int totalScore;
        int perMatchScore;
        float average;

        Batsman (string n, int m, int score, int matchScore)
        {
            name = n;
            matches = m;
            totalScore = score;
            perMatchScore = matchScore;
        }
        void calculateAverage()
        {
            average = totalScore / matches;
        }
        void display()
        {
            cout << "Name: " << name << endl;
            cout << "Matches: " << matches << endl;
            cout << "totalScore: " << totalScore << endl;
            cout << "perMatchScore: " << perMatchScore <<
                endl;
            cout << "Average: " << average << endl << endl;
        }
};

class Bowler : public Player {
    public:
        int noOfWickets;

        Bowler (string n, int m, int wickets)
        {
            name = n;
            matches = m;
            noOfWickets = wickets;
        }
        void display()
        {
            cout << "Name: " << name << endl;
            cout << "Matches: " << matches << endl;
            cout << "noOfWickets: " << noOfWickets << endl <<
                endl;
        }
};

int main ()
{
    Batsman bat1("Batsman 1", 10, 200, 5);
    Bowler bow1("Bowler 1", 10, 2);

    bat1.calculateAverage();
    bat1.display();
    bow1.display();

    return 0;
}
```

Output:

```
/tmp/18qXqNKa1d.o
Name: Batsman 1
Matches: 10
totalScore: 200
perMatchScore: 5
Average: 20

Name: Bowler 1
Matches: 10
noOfWickets: 2


=== Code Execution Successful ===
```