NAME : MUHAMMAD ZAIN

SECTION: BS-AI-IIA

ROLL NO: 052

## OOPS ASSIGNMENT NO 3

# PROGRAM NO 1

#include <iostream>

#include <string>

using namespace std;

class Publication {

protected:

string title;

float price;

public:

void getData() {

cout << "Enter the title: ";

getline(cin, title);

cout << "Enter the price: ";

cin >> price;

cin.ignore(); // To clear the newline character from the input buffer

}

void putData() {

cout << "Title: " << title << endl;

cout << "Price: " << price << endl;

}

};

class Book : public Publication {

private:

int pageCount;

public:

void getData() {

Publication::getData();

cout << "Enter the page count: ";

cin >> pageCount;

cin.ignore(); // To clear the newline character from the input buffer

}

void putData() {

Publication::putData();

cout << "Page Count: " << pageCount << endl;

}

};

class Tape : public Publication {

private:

float playingTime;

public:

void getData() {

Publication::getData();

cout << "Enter the playing time (in minutes): ";

cin >> playingTime;

cin.ignore(); // To clear the newline character from the input buffer

}

void putData() {

Publication::putData();

cout << "Playing Time: " << playingTime << " minutes" << endl;

}

};

int main() {

Book;

Tape;

cout << "Enter details for the book:\n";

book.getData();

cout << "\nEnter details for the tape:\n";

tape.getData();

cout << "\nBook details:\n";

book.putData();

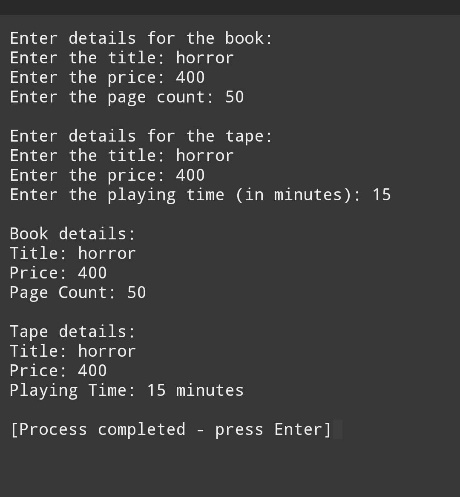
cout << "\nTape details:\n";

tape.putData();

return 0;

}

**Output**



# PROGRAM NO 2

#include <iostream>

using namespace std;

class Sales {

protected:

float sales[3];

public:

void getData() {

cout << "Enter sales for last three months: ";

cin >> sales[0] >> sales[1] >> sales[2];

}

void putData() {

cout << "Sales for last three months: ";

cout << sales[0] << " " << sales[1] << " " << sales[2] << endl;

}

};

class Publication {

protected:

string title;

public:

void getData() {

cout << "Enter title: ";

cin.ignore();

getline(cin, title);

}

void putData() {

cout << "Title: " << title << endl;

}

};

class Book : public Publication, public Sales {

public:

void getData() {

Publication::getData();

Sales::getData();

}

void putData() {

Publication::putData();

Sales::putData();

}

};

class Tape : public Publication, public Sales {

public:

void getData() {

Publication::getData();

Sales::getData();

}

void putData() {

Publication::putData();

Sales::putData();

}

};

int main() {

Book;

Tape;

book.getData();

tape.getData();

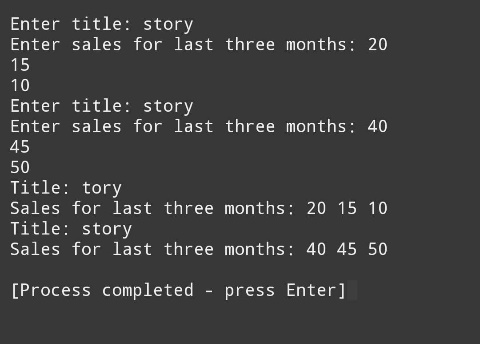
book.putData();

tape.putData();

return 0;

}

## Output



`

# PROGRAM NO 3

## #include <iostream>

## using namespace std;

## class Sales {

## protected:

## float sales[3];

## public:

## void getData() {

## cout << "Enter sales for last three months: ";

## cin >> sales[0] >> sales[1] >> sales[2];

## }

## void putData() {

## cout << "Sales for last three months: ";

## cout << sales[0] << " " << sales[1] << " " << sales[2] << endl;

## }

## };

## class Publication {

## protected:

## string title;

## public:

## void getData() {

## cout << "Enter title: ";

## cin.ignore();

## getline(cin, title);

## }

## void putData() {

## cout << "Title: " << title << endl;

## }

## };

## class Book : public Publication, public Sales {

## public:

## void getData() {

## Publication::getData();

## Sales::getData();

## }

## void putData() {

## Publication::putData();

## Sales::putData();

## }

## };

## class Tape : public Publication, public Sales {

## public:

## void getData() {

## Publication::getData();

## Sales::getData();

## }

## void putData() {

## Publication::putData();

## Sales::putData();

## }

## };

## int main() {

## Book;

## Tape;

## 

## book.getData();

## tape.getData();

## 

## book.putData();

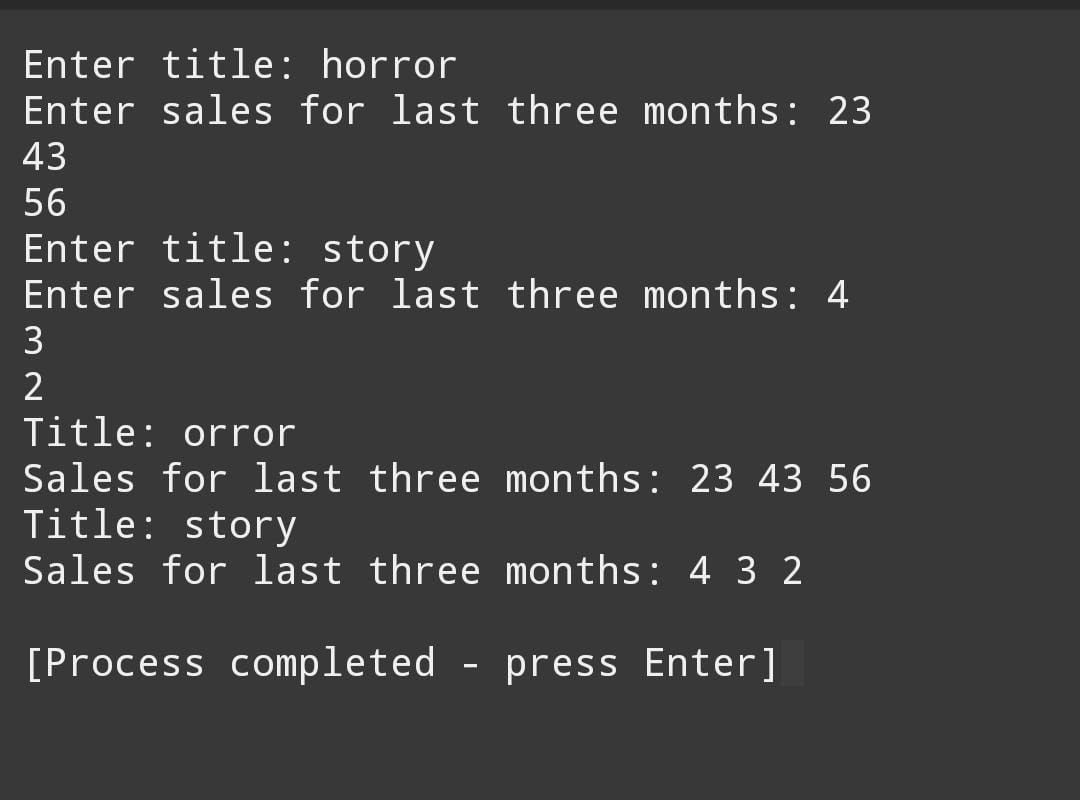
## tape.putData();

## 

## return 0;

## }

## OUTPUT



# PROGRAM NO 4

#include <iostream>

using namespace std;

enum Period { HOURLY, WEEKLY, MONTHLY };

class Employee {

protected:

string name;

int id;

static int nextId;

public:

Employee() : name(""), id(nextId++) {}

Employee(string n) : name(n), id(nextId++) {}

string getName() { return name; }

int getId() { return id; }

};

int Employee::nextId = 1;

class Employee2 : public Employee {

protected:

double compensation;

Period;

public:

void getData() {

cout << "Enter compensation: ";

cin >> compensation;

cout << "Enter period (0 for HOURLY, 1 for WEEKLY, 2 for MONTHLY): ";

int p;

cin >> p;

switch (p) {

case 0:

period = HOURLY;

break;

case 1:

period = WEEKLY;

break;

case 2:

period = MONTHLY;

break;

default:

cout << "Invalid period!" << endl;

}

}

void putData() {

cout << "Name: " << getName() << endl;

cout << "ID: " << getId() << endl;

cout << "Compensation: " << compensation << endl;

cout << "Period: " << (period == HOURLY ? "HOURLY" : period == WEEKLY ? "WEEKLY" : "MONTHLY") << endl;

}

};

class Manager : public Employee2 {

public:

void getData() {

Employee2::getData();

// Add manager-specific data

}

void putData() {

Employee2::putData();

// Add manager-specific data

}

};

class Scientist : public Employee2 {

public:

void getData() {

Employee2::getData();

// Add scientist-specific data

}

void putData() {

Employee2::putData();

// Add scientist-specific data

}

};

class Laborer : public Employee2 {

public:

void getData() {

Employee2::getData();

// Add laborer-specific data

}

void putData() {

Employee2::putData();

// Add laborer-specific data

}

};

int main() {

Manager;

Scientist;

Laborer;

manager.getData();

scientist.getData();

laborer.getData();

manager.putData();

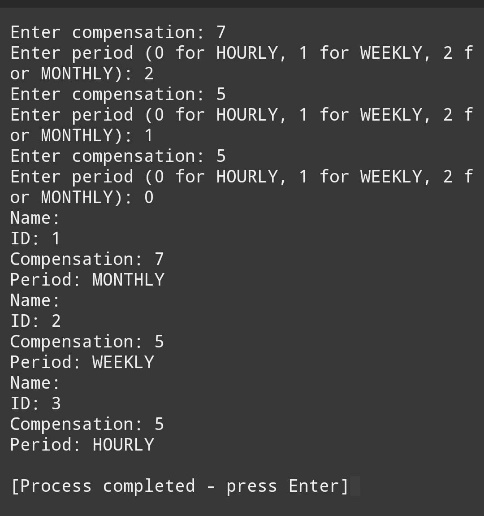
scientist.putData();

laborer.putData();

return 0;

}

**output**



# PROGRAM NO 5

# #include <iostream>

# #include <string>

# #include <cmath>

# using namespace std;

# class Shape {

# protected:

# string color;

# public:

# Shape(const string& c) : color(c) {}

# void printColor() {

# cout << "Color: " << color << endl;

# }

# };

# class Circle : public Shape {

# private:

# double radius;

# public:

# Circle(const string& c, double r) : Shape(c), radius(r) {}

# double calculateArea() {

# return M\_PI \* radius \* radius;

# }

# void printArea() {

# cout << "Area of the circle: " << calculateArea() << endl;

# }

# };

# class Rectangle : public Shape {

# private:

# double length;

# double width;

# public:

# Rectangle(const string& c, double l, double w) : Shape(c), length(l), width(w) {}

# double calculateArea() {

# return length \* width;

# }

# void printArea() {

# cout << "Area of the rectangle: " << calculateArea() << endl;

# }

# };

# int main() {

# Circle circle("Red", 5.0);

# circle.printColor();

# circle.printArea();

# Rectangle rectangle("Blue", 4.0, 6.0);

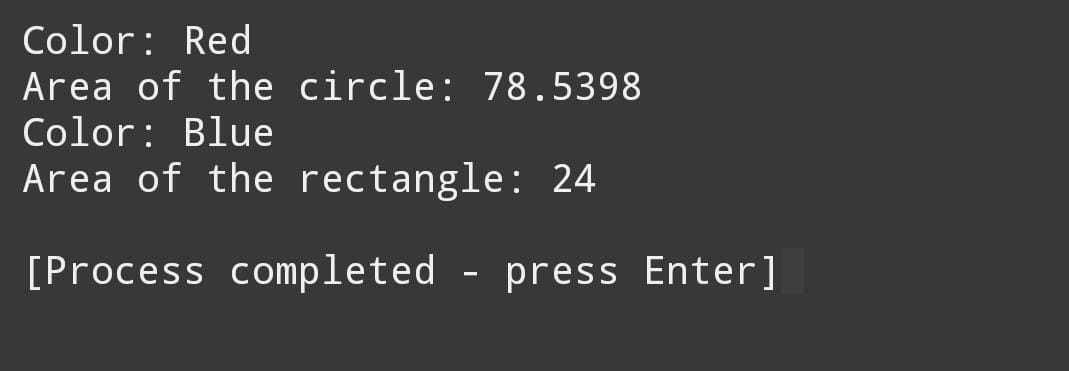
# rectangle.printColor();

# rectangle.printArea();

# return 0;

# }

# Output



# PROGRAM NO 6

# #include <iostream>

# #include <string>

# using namespace std;

# class Employee {

# protected:

# string name;

# int employeeID;

# string department;

# public:

# Employee(string n, int id, string dept) : name(n), employeeID(id), department(dept) {}

# string getName() const { return name; }

# int getEmployeeID() const { return employeeID; }

# string getDepartment() const { return department; }

# void setName(const string& n) { name = n; }

# void setEmployeeID(int id) { employeeID = id; }

# void setDepartment(const string& dept) { department = dept; }

# };

# class SalariedEmployee : public Employee {

# private:

# double annualSalary;

# public:

# SalariedEmployee(string n, int id, string dept, double salary) : Employee(n, id, dept), annualSalary(salary) {}

# double getAnnualSalary() const { return annualSalary; }

# double calculateMonthlyPay() const { return annualSalary / 12; }

# void setAnnualSalary(double salary) { annualSalary = salary; }

# };

# class CommissionEmployee : public Employee {

# private:

# double sales;

# double commissionRate;

# public:

# CommissionEmployee(string n, int id, string dept, double sales, double rate) : Employee(n, id, dept), sales(sales), commissionRate(rate) {}

# double getSales() const { return sales; }

# double getCommissionRate() const { return commissionRate; }

# double calculateTotalPay() const { return sales \* commissionRate; }

# void setSales(double s) { sales = s; }

# void setCommissionRate(double rate) { commissionRate = rate; }

# };

# int main() {

# SalariedEmployee salariedEmp("John Doe", 101, "Sales", 50000.0);

# CommissionEmployee commissionEmp("Jane Smith", 102, "Marketing", 10000.0, 0.1);

# cout << "Salaried Employee: " << salariedEmp.getName() << endl;

# cout << "Monthly Pay: " << salariedEmp.calculateMonthlyPay() << endl;

# cout << "Commission Employee: " << commissionEmp.getName() << endl;

# cout << "Total Pay: " << commissionEmp.calculateTotalPay() << endl;

# return 0;

# }

# Output

