**LAB 06**

**QUESTION 1:**

**Create a base class named Rectangle that includes data members for the length and width of a Rectangle, as well as functions to assign and display those values. Derive a class named Block that contains an additional data member to store height, and contains functions to assign and display the height. Write a main() function that demonstrates the classes by instantiating and displaying the values for both a Rectangle and a Block..**

**PROGRAM**:

#include <iostream>

using namespace std;

class Rectangle {

    private:

        int length;

        int width;

    public:

        void setLength (int l) {

            length = l;

        }

        int getLength () {

            return length;

        }

        void setWidth (int *w*) {

            width = w;

        }

        int getWidth (){

            return width;

        }

};

class Block : public Rectangle{

    private:

        int height;

    public:

        void setHeight (int *h*) {

            height = h;

        }

        int getHeight() {

            return height;

        }

};

int main() {

    Rectangle r;

    Block b;

    r.setLength(5);

    r.setWidth(10);

    cout << "Rectangle's length is " << r.getLength() << "and width is " << r.getWidth() << endl << endl;

    b.setHeight(4);

    b.setLength(6);

    b.setWidth(8);

    cout << "Block's length is " << b.getLength() << ", width is " << b.getWidth() << "and height is " << b.getHeight() << endl;

    return 0;

}

**RESULT:**

**A black background with white text

AI-generated content may be incorrect.**

**QUESTION#2**

**Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a getdata() function to get its data from the user at the keyboard, and a putdata() function to display its data.Write a main() program to test the book and tape classes by creating instances of them, asking the user to fill in data with getdata(), and then displaying the data with putdata().**

**PROGRAM**:

#include <iostream>

using namespace std;

class publication{

    protected:

    string title;

    float price;

    public:

    void getdata(){

        cout<<"Enter book title :";

        cin>>title;

        cout<<"Enter book price :";

        cin>>price;

    }

    void putdata(){

        cout<<"title :"<<title<<endl;

        cout<<"price :"<<price<<endl;

    }

};

class book : public publication{

 private:

 int count;

 public:

 void getdata(){

     publication::getdata();

     cout << "Enter page count: ";

     cin>>count;

 }

 int  getbookcount(){

  return count;

 }

 void putdata(){

    publication::putdata();

    cout<<"page count :"<<getbookcount()<<endl;

 }

};

class tape: public publication{

private:

float playtime;

public:

void getdata(){

    publication::getdata();

    cout << "Enter playing time: ";

    cin>>playtime;

}

float getplaytime(){

    return playtime;

}

 void putdata(){

    publication::putdata();

    cout<<"playtime :"<<getplaytime()<<endl;

 }

};

int main() {

    cout<<"-----book class------"<<endl;

    book b1;

    cout<<"Input"<<endl;

    b1.getdata();

    cout<<"Ouput"<<endl;

    b1.putdata();

    cout<<"-----tape class------"<<endl;

    tape t1;

    cout<<"Input"<<endl;

    t1.getdata();

    cout<<"Output"<<endl;

    t1.putdata();

    return 0;

}

A screenshot of a computer program

AI-generated content may be incorrect.

**QUESTION#3**

**A Create a class Person with default constructor. Inherit a new class called Student from class Person with default constructor and create a Teacher class which inherit from Person that also contain default constructor. In the main function create Teacher,Student and Person objects and observe the results.**

#include <iostream>

using namespace std;

class person{

    public:

    person(){

        cout<<"person class called"<<endl;

    }

};

class Teacher : public person{

    public:

    Teacher(){

        cout<<"teacher class called"<<endl;

    }

};

class student : public person{

    public:

    student(){

        cout<<"student class called"<<endl;

    }

};

int main() {

     person p1;

     Teacher t1;

     student s1;

    return 0;

}

A screen shot of a computer screen

AI-generated content may be incorrect.

**QUESTION#4**

**A Design a class named Employee. The class should keep the following information in**

** Employee name**

** Employee number**

** Hire date**

**Write one or more constructors and the appropriate accessor and mutator functions for the class.Next, write a class named ProductionWorker that is derived from the Employee class.The ProductionWorker class should have member variables to hold the following information:**

** Shift (an integer)**

** Hourly pay rate (a double )**

**The workday is divided into two shifts: day and night. The shift variable will hold an integer value representing the shift that the employee works. The day shift is shift 1, and the night shift is shift 2. Write one or more constructors and the appropriate accessor and mutator functions for the class. Demonstrate the classes by writing a program that uses a ProductionWorker object.**

#include <iostream>

using namespace std;

class employee{

  private:

   string name;

   int number;

   string hiredate;

  public:

  employee(string *name*,int *number*,string *hiredate*){

       this->name=*name*;

       this->number=*number*;

       this->hiredate=*hiredate*;

  }

  string getname(){ return name;}

  string gethiredate(){ return hiredate;}

  int getnumber(){ return number;}

  void setname(string *name*){

    this->name=*name*;

  }

  void sethiredate(string *hiredate*){

    this->hiredate=*hiredate*;

  }

  void setnumber(int *number*){

    this->number=*number*;

  }

  void display(){

    cout<<"name :"<<getname()<<endl;

    cout<<"hire date: "<<gethiredate()<<endl;

    cout<<"number: "<<getnumber()<<endl;

  }

};

class productionWorker:public employee{

  private:

  int shift;

  double pay;

  public:

  productionWorker(string *name*,int *number*,string *hiredate*,int *shift*,double *pay*): employee(*name*,*number*,*hiredate*){

    this->shift=*shift*;

    this->pay=*pay*;

  }

int getshift(){ return shift;}

double getpay(){ return pay;}

void setshift(int *shift*){

  this->shift=*shift*;

}

void setpay(double *pay*){

  this->pay=*pay*;

}

void displayWorker(){

  display();

  if(getshift()==1){

  cout<<"shift :"<<"morning"<<endl;

  }

  else{

    cout<<"shift :"<<"night"<<endl;

  }

  cout<<"pay :"<<getpay()<<endl;

}

};

int main() {

  int shift;

  cout<<"Enter shift of worker 1(Morning) and 2(Night):";

  cin>>shift;

  while(shift!=1 && shift !=2){

    cout<<"invalid shift enter again :";

    cin>>shift;

  }

  productionWorker p1("Kali",23,"23-2-2025",shift,2000);

  p1.displayWorker();

    return 0;

}

A screenshot of a computer

AI-generated content may be incorrect.

**QUESTION#5**

**For In a vehicle manufacturing company, there are different types of vehicles:**

** Vehicle class that contains general attributes like manufacturer, model, and methods like getInfo().**

** Car class that inherits from Vehicle, with additional attributes like numDoors and methods like getCarDetails().**

** ElectricCar class inherits from Car, with additional attributes like batteryCapacity, and a method getElectricCarDetails().**

**Write a program to implement this hierarchy and demonstrate how a ElectricCar object can access data and methods from both Vehicle and Car classes. Include methods to input and output the details of the electric car.**

#include <iostream>

using namespace std;

class vehicle {

    protected:

        string manufacture;

        string model;

    public:

        vehicle(string *manufacture*,string *model*){

            this->manufacture=*manufacture*;

            this->model=*model*;

        }

        void getinfo(){

            cout<<"manufacture is :"<<manufacture<<endl;

            cout<<"model is :"<<model<<endl;

        }

};

class car : public vehicle {

    protected:

    int numDoors;

      public:

       car(string *manufacture*,string *model*,int *numDoors*):vehicle(*manufacture*,*model*){

       this->numDoors=*numDoors*;

       }

        void getCardetails(){

            cout<<"manufacture is :"<<manufacture<<endl;

            cout<<"model is :"<<model<<endl;

            cout<<"num of doors is :"<<numDoors<<endl;

        }

};

class ElectricCar : public car {

    private:

    int batteryCapacity;

       public:

       ElectricCar(string *manufacture*,string *model*,int *numDoors*,int *batteryCapacity*): car(*manufacture*,*model*,*numDoors*){

       this->batteryCapacity=*batteryCapacity*;

       }

    int getbatteryCapacity(){

             return batteryCapacity;

        }

    void getElectricCardetails(){

            cout<<"manufacture is :"<<manufacture<<endl;

            cout<<"model is :"<<model<<endl;

            cout<<"num of doors is :"<<numDoors<<endl;

            cout<<"capacity of battery is :"<<batteryCapacity<<endl;

        }

};

int main(){

    ElectricCar E("Tesla", "Model S", 4, 100);

    cout << "Electric Car Details:" << endl;

    E.getElectricCardetails();

}

A screenshot of a computer screen

AI-generated content may be incorrect.

**QUESTION#6**

**In a university, the base class is Person which includes general details like name and age.**

** The Teacher class inherits from Person and has attributes like subject and salary, with a method getTeacherDetails().**

** The Student class also inherits from Person and has attributes like rollNumber and course, with a method getStudentDetails().**

**Write a program where Person is the base class, and both Teacher and Student inherit from it. Create a Teacher object and a Student object, input their details, and display them using their respective methods. Demonstrate how both derived classes share common properties from the Person class.**

#include <iostream>

using namespace std;

class person{

    protected:

    int age;

    string name;

    public:

    person(string name,int age):name(name),age(age){}

    void display() {

        cout << "Name: " << name << endl;

        cout << "Age: " << age << endl;

    }

};

class Teacher : public person{

    private:

    int salary;

    string subject;

    public:

    Teacher(string name,int age,int salary,string subject) : person(name,age){

        this->salary=salary;

        this->subject=subject;

    }

    void getTeacherDetails(){

        display();

        cout<<"salary: "<<salary<<endl;

        cout<<"subject: "<<subject<<endl;

    }

};

class student : public person{

    private:

    int rollNo;

    string course;

    public:

    student(string name,int age,int rollNo,string course) : person(name,age){

        this->rollNo=rollNo;

        this->course=course;

    }

    void getstudentDetails(){

        display();

        cout<<"roll number: "<<rollNo<<endl;

        cout<<"course: "<<course<<endl;

    }

};

int main() {

     Teacher t1("Ahmed",35,200000,"Programming fundamentals");

     cout <<endl<<"--- Teacher Details ---"<<endl;

     t1.getTeacherDetails();

     cout <<endl<<"--- Student Details ---"<<endl;

     student s1("Umais",20,24122,"Cyber Security");

     s1.getstudentDetails();

    return 0;

}

A screenshot of a computer

AI-generated content may be incorrect.