VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_MCQ

Attempt : 2
Total Mark : 20
Marks Obtained : 20

Section 1 : MCQ

1. What will be the output of the following code?

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

class Shape {
public:
   virtual void display() {
      cout << "Shape! ";
   }
};

class Square : public Shape {
   int s = 10, area;
public:
   Square(int side){
      s += side;
   }
   void display() override {
      Shape::display();
      area = s * s;
      cout << area << " ";
   }
};

int main() {
   Square s(5);
   Shape* ptr = &s;
   ptr->display();
   s.display();
   return 0;
}
```

Answer

Shape! 225 Shape! 225

Status : Correct ***Marks : 1/1***

2. What will be the output of the following program?

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

class Complex {
public:
   int a;
   int b;
   Complex(int r, int i) {
      a = --r;
      b = --i;
   }
   Complex operator+() {
      return Complex(a, b);
```

```
    }
};

int main() {
    Complex a(-3, 4);
    Complex c = +a;
    cout << c.a << " " << c.b;
    return 0;
}
```

Answer

-5 2

Status : Correct ***Marks : 1/1***

3. What is the output for the following code?

```
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

double magnitude(double x, double y, double z) {
    return sqrt(x * x + y * y + z * z);
}

double magnitude(double x1, double x2) {
    return sqrt(x1 * x2 + x1 / x2);
}

int main() {
    cout << fixed << setprecision(2);
    cout << magnitude(1.0, 2.0, 3.0) << endl;
    cout << magnitude(4.0, 6.0);
    return 0;
}
```

Answer

3.744.97

Status : Correct *Marks : 1/1*

4. What will be the output of the following code?

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

class Animal {
public:
   virtual void sound() {
      cout << "Animal sound! ";
   }
};

class Dog : public Animal {
   int age, tag;
public:
   Dog(){
      age = 2;
      tag = 12345;
   }
   void sound() override {
      cout << age << " " << tag << " ";
      Animal::sound();
   }
};

int main() {
   Dog m;
   Animal* ptr = &m;
   m.sound();
   ptr->sound();
   return 0;
}
```

Answer

2 12345 Animal sound! 2 12345 Animal sound!

Status : Correct *Marks : 1/1*

5. What will be the output of the following code?

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

class Rain {
public:
    virtual void func() = 0;
    void fun() {
        cout << "Pouring ";
    }
};

class Drizzle : public Rain {
public:
    void func() override {
        cout << "Light ";
    }

    void func(int x) {
        if (x < 2) {
            cout << "Drizzle ";
            Rain::fun();
        }
        else {
            Rain::fun();
            cout << "Moderate ";
        }
    }
};

int main() {
    Drizzle d;
    d.func();
    d.func(1);
    d.func(3);
    return 0;
}
```

Answer

Light Drizzle Pouring Pouring Moderate

Status : Correct *Marks : 1/1*

6. What is the output for the following code?

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

int globalValue = 150;

class ScopeHiding {
public:
   int globalValue;
};

int main() {
   ScopeHiding obj;
   obj.globalValue = 50;
   cout << obj.globalValue << endl;
   cout << globalValue;
   return 0;
}
```

Answer

50150

Status : Correct *Marks : 1/1*

7. What will be the output for the following program?

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

class Animal {
public:
   virtual void makeSound() const = 0;
};
```

```
class Dog : public Animal {
public:
    void makeSound() const override {
        cout << "Woof Woof!" << endl;
    }
};

class Cat : public Animal {
public:
    void makeSound() const override {
        cout << "Meow!" << endl;
    }
};


int main() {
    Animal* animal1 = new Dog();
    Animal* animal2 = new Cat();

    animal1->makeSound();
    animal2->makeSound();

    return 0;
}
```

Answer

Woof Woof!Meow!

Status : Correct ***Marks : 1/1***

8. What will be the output of the following code?

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

class Flight {
public:
    virtual void V() = 0;
    void fun() {
        cout << "Started ";
    }
};

class Fly : public Flight {
public:
    void V() override {
        cout << "Accelerating ";
    }

    void fun() {
        Flight::fun();
        cout << "Flying ";
    }
};

int main() {
    Fly f;
    f.fun();
    f.V();
    return 0;
}
```

Answer

Started Flying Accelerating

Status : Correct ***Marks : 1/1***

9. What will be the output of the following code?

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

class Base {
public:
    virtual void func() {
        cout << "Base ";
    }
};
```

```
class Derived : public Base {
public:
   void func() {
      cout << "Derived ";
      Base::func();
   }
};

int main() {
   Derived d;
   Base* b = &d;
   b->func();
   return 0;
}
```

Answer

Derived Base

Status : Correct ***Marks : 1/1***

10. What will be the output for the following program?

Note: This kind of question will be helpful in clearing TCS tests.

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

class Media {
public:
   virtual void play() const = 0;
};

class Audio : public Media {
public:
   void play() const override {
      cout << "Audio Playing..." << endl;
   }
};
```

```
class Video : public Media {
public:
   void play() const override {
      cout << "Video Playing..." << endl;
   }
};

int main() {
   Media* media1 = new Audio();
   Media* media2 = new Video();

   media1->play();
   media2->play();

   return 0;
}
```

Answer

Audio Playing...Video Playing...

Status : Correct ***Marks : 1/1***

11. In the given code snippet, which class or classes are abstract classes?

```
class Hill {
public:
   virtual void climb() = 0;
};

class Jack {
public:
   virtual void jump() = 0;
};

class Jill : public Hill, public Jack {
public:
   void climb() override {}
   void jump() override {}
};
```

Answer

Hill and Jack are abstract classes

Status : Correct ***Marks : 1/1***

12. What will be the output of the following code?

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

class Bool {
public:
   bool value;
   Bool(bool v) {
      value = v;
   }

   bool operator!() {
      return !value;
   }
};

int main() {
   Bool myTrue(true);
   Bool myFalse(false);
   cout << !myTrue.value << " " << !myFalse.value;
   return 0;
}
```

Answer

0 1

Status : Correct ***Marks : 1/1***

13. What will be the output of the following program?

```
#include <iostream>
```

```
using namespace std;

class Number {
public:
    int x, y, z;

    Number(int a, int b, int c) {
        x = a;
        y = b;
        z = c;
    }

    Number operator++() {
        ++x;
        ++y;
        ++z;
        return *this;
    }
};

int main() {
    Number p(-99, 10, 29);
    Number q = ++p;

    cout << p.x << " " << p.y << " " << p.z << endl;
    cout << q.x << " " << q.y << " " << q.z;

    return 0;
}
```

Answer

```
-98 11 30-98 11 30
```

Status : Correct ***Marks : 1/1***

14. What will be the output of the following code?

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

```
class Crystal {
public:
   virtual void Fun() = 0;
   void Fun(int x) {
      cout << (x < 3 ? "Average " : "Excellent ");
   }
};

class Shine : public Crystal {
public:
   void Fun() override {}
   void Fun(int y) {
      Crystal::Fun(5);
      cout << (y < 2 ? "Low " : "High ");
   }
};

int main() {
   Shine obj;
   obj.Fun();
   obj.Fun(4);
   obj.Fun(1);
   return 0;
}
```

Answer

Excellent High Excellent Low

Status : Correct ***Marks : 1/1***

15. What is the output of the following program?

```
#include <iostream>

class Shape {
public:
   virtual void calculateArea() = 0;
};
```

```
class Rectangle : public Shape {
public:
   void calculateArea() {
      std::cout << "Calculating area of Rectangle" << std::endl;
   }
};

class Circle : public Shape {
public:
   void calculateArea() {
      std::cout << "Calculating area of Circle" << std::endl;
   }
};

int main() {
   Shape* shapePtr;
   Rectangle rectangle;
   Circle circle;

   shapePtr = &rectangle;
   shapePtr->calculateArea();

   shapePtr = &circle;
   shapePtr->calculateArea();

   return 0;
}
```

Answer

Calculating area of Rectangle followed by Calculating area of Circle

Status : Correct ***Marks : 1/1***

16. What is the output for the following code?

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

```
char mixColors(char color1) {
   return color1 + 2;
}

char mixColors(char color1, char color2) {
   if (color1 == 'r' && color2 == 'b')
      return 'p';
   else if (color1 == 'b' && color2 == 'y')
      return 'g';
   else if (color1 == 'r' && color2 == 'y')
      return 'o';
   else
      return 'u';
}

int main() {
   cout << mixColors('r') << endl;
   cout << mixColors('b', 'y');
   return 0;
}
```

Answer

tg

Status : Correct ***Marks : 1/1***

17. What will be the output of the following code?

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

class Base {
public:
   virtual int calc(int a, int b) {
      return a - b;
   }
};

class Derived : public Base {
```

```
public:
    int calc(int a, int b) override {
        return a % b;
    }
};

int main() {
    Base* ptr;
    Derived d;
    ptr = &d;
    int result = ptr->calc(10, 4) * d.calc(7, 8) - ptr->calc(5, 1);
    cout << result;
    return 0;
}
```

Answer

14

Status : Correct ***Marks : 1/1***

18. What will be the output of the following code?

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

class Number {
public:
    int value;

    Number(int v) {
        value = v;
    }

    Number operator-() {
        return Number(--value);
    }

    Number operator+() {
        return Number(++value);
    }
};

int main() {
    Number m(-5);
    Number n = -m;
    Number p = +n;
    cout << m.value << " " << n.value << " " << p.value;
    return 0;
}
```

Answer

-6 -5 -5

Status : Correct ***Marks : 1/1***

19. What is the output for the following code?

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

void incrementByOne(int value) {
    int result = value + 1;
    cout << result << endl;
}

void incrementByOne(double value) {
    double result = value + 1.0;
    cout << result;
}

int main() {
    incrementByOne(7);
    incrementByOne(5.5);
    return 0;
}
```

Answer

86.5

Status : Correct ***Marks : 1/1***

20. What will be the output of the following code?

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

class Vehicle {
public:
   virtual void start() {
      cout << "Virtual ";
   }
};

class Car : public Vehicle {
public:
   Car(){
      wheels = 4;
   }

   Car(int n){
      wheels = n;
   }
   void start() override{
      cout << wheels << " wheels ";
   }

private:
   int wheels;
};

int main() {
   Car c(6);
   Vehicle* v = &c;
   v->start();
   return 0;
}
```

Answer

6 wheels

Status : Correct ***Marks : 1/1***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Easy_Function Overloading

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Teju is working on a unit converter application that can convert lengths from millimetres to centimetres and kilometres. The application uses function overloading to handle the different unit types.

She wants to implement a UnitConverter class that contains two overloaded functions for conversion:

double convert(double millimetres) - If the unit type is 1, convert the value to centimetres.double convert(int unitType, double value) - If the unit type is 2, convert the value to kilometres.

Assist Teju in completing the converter application.

Answer

```
// You are using GCC
#include<bits/stdc++.h>
using namespace std;

class UnitConverter
{
   public:
   double convert(double millimetres)
   {
      return millimetres/10.0;
   }
   double convert(int unitType,double millimetres)
   {
      return millimetres/1000000.0;
   }

};
int main()
{
   int unitType;
   double N;
   cin>>unitType>>N;

   UnitConverter c;
   cout<<fixed<<setprecision(2);
   if(unitType==1)
   {
      cout<<c.convert(N)<<"cm";
   }
   else if(unitType==2)
   {
     cout<<c.convert(unitType,N)<<"km";
   }
   else
   {
      cout<<"Invalid unit type!";
   }
   return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Vamsi is a young and curious student who is eager to learn about finding the minimum number among a set of integers and double values. He needs a program to find the minimum number from a given set of values.

Help him solve the program by overloading the function named findMin.

Answer

```
// You are using GCC
#include<bits/stdc++.h>
using namespace std;

int findMin(int a,int b, int c)
{
   int m=a;
   if(b<m)m=b;
   if(c<m)m=c;
   return m;
}

double findMin(double a,double b,double c)
{
   double m=a;
   if(b<m)m=b;
   if(c<m)m=c;
   return m;
}

int main()
{
   int int1,int2,int3;
   double d1,d2,d3;

   cin >> int1 >> int2 >> int3;
   cin >> d1 >> d2 >> d3;

   int minInt=findMin(int1,int2,int3);
   double minDouble=findMin(d1,d2,d3);

   cout<<"Minimum integer: "<<minInt<<endl;
```

```
    cout<<"Minimum double-point value: "<<fixed<<setprecision(2)<<minDouble;

    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Easy_Operator Overloading

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Implement a Fraction class that represents a fraction with a numerator and a denominator. Overload the '+' operator to add two fractions and return the result as a simplified fraction.

Function Specifications: Fraction operator+(const Fraction&amp; other) const

Answer

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

class Fraction {
```

```
public:
   int num, den;
   Fraction(int n, int d) : num(n), den(d) {}

   int gcd(int a, int b) const {
      return b == 0 ? a : gcd(b, a % b);
   }

   Fraction operator+(const Fraction& other) const {
      int n = num * other.den + other.num * den;
      int d = den * other.den;
      int g = gcd(n, d);
      return Fraction(n / g, d / g);
   }

   void display() const {
      cout << num << "/" << den << endl;
   }
};

int main() {
   int n1, d1, n2, d2;
   cin >> n1 >> d1 >> n2 >> d2;

   Fraction f1(n1, d1), f2(n2, d2);
   (f1 + f2).display();

   return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

John is driving a car with an initial velocity (in m/s) that suddenly accelerates at a constant rate (in m/s2) for a certain time (in seconds). He wants to write a program that calculates and displays the final velocity of the car.

Help John calculate the final velocity by overloading the * operator in the

Acceleration class.

Formula: Final velocity = Initial velocity + (Acceleration * time)

Answer

```
// You are using GCC
#include <iostream>
#include <iomanip>
using namespace std;

class Acceleration {
public:
   float a;
   Acceleration(float acc) : a(acc) {}

   float operator*(float t) const {
      return a * t;
   }
};

int main() {
   float v, a, t;
   cin >> v >> a >> t;

   Acceleration acc(a);
   cout << fixed << setprecision(1) << (v + (acc * t)) << " m/s" << endl;

   return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Easy_Virtual Functions

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Imagine you are creating a SleepTracker program that demonstrates the use of inheritance and virtual functions. The program helps users analyze their sleep patterns, distinguishing between weekday and weekend sleep durations.

Implement a base class called SleepTracker with attributes for bedtime and wakeup times and virtual functions for input and duration calculation.

Derive two subclasses, WeekdaySleep and WeekendSleep, inheriting from SleepTracker. Override the virtual functions for specific behavior on weekdays and weekends.

Answer

```
// You are using GCC
#include <iostream>
using namespace std;

class SleepTracker {
protected:
    int bh, bm, wh, wm;
public:
    virtual void input(int h1, int m1, int h2, int m2) {
        bh = h1, bm = m1, wh = h2, wm = m2;
    }
    virtual int duration() const {
        int m = (wh * 60 + wm) - (bh * 60 + bm);
        if (m < 0) m += 1440;
        return m;
    }
    virtual ~SleepTracker() {}
};

class WeekdaySleep : public SleepTracker {
public:
    void input(int h1, int m1, int h2, int m2) override {
        SleepTracker::input(h1, m1, h2, m2);
    }
    int duration() const override {
        return SleepTracker::duration();
    }
};

class WeekendSleep : public SleepTracker {
public:
    void input(int h1, int m1, int h2, int m2) override {
        SleepTracker::input(h1, m1, h2, m2);
    }
    int duration() const override {
        return SleepTracker::duration();
    }
};

int main() {
    int bh1, bm1, wh1, wm1, bh2, bm2, wh2, wm2;
    cin >> bh1 >> bm1 >> wh1 >> wm1 >> bh2 >> bm2 >> wh2 >> wm2;
    WeekdaySleep wd;
    WeekendSleep we;

    wd.input(bh1, bm1, wh1, wm1);
    we.input(bh2, bm2, wh2, wm2);

    int wd_m = wd.duration(), we_m = we.duration();
    cout << "Weekday: " << wd_m / 60 << "h " << wd_m % 60 << "m\n";
    cout << "Weekend: " << we_m / 60 << "h " << we_m % 60 << "m\n";

    if (wd_m > we_m) cout << "User slept more on weekdays.\n";
    else if (we_m > wd_m) cout << "User slept more on the weekend.\n";
    else cout << "User slept the same amount on weekdays and weekend.\n";

    return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Renu works for a retail store that sells two types of items: wooden items and electronics. The store needs a program to calculate the total amount for a customer's purchase based on their choice of item type and the quantity or cost of the item(s).

Create a base class, ItemType, with one virtual function.

virtual double calculateAmount()

Create a class called wooden that extends ItemType with a number of items and cost as its private attributes. Obtain the data members and override the virtual function.

amount = number of items * cost

Create a class for electronics that extends ItemType with cost as its private attribute. Obtain the data member and override the virtual function.

amount = 80% of the amount (20% discount)

In the main method, obtain a choice.

If the choice is 1, create an object for the wooden class and call the method.

If the choice is 2, create an object for the electronics class and call the method.

Answer

```
// You are using GCC
#include <iostream>
#include <iomanip>
using namespace std;

class ItemType {
public:
   virtual double calculateAmount() = 0;
   virtual ~ItemType() {}
};

class Wooden : public ItemType {
   int n;
   double c;
public:
   Wooden(int num, double cost) : n(num), c(cost) {}
   double calculateAmount() override {
      return n * c;
   }
};

class Electronics : public ItemType {
   double c;
public:
   Electronics(double cost) : c(cost) {}
   double calculateAmount() override {
      return c * 0.8;
   }
};

int main() {
   int ch;
```

```
    cin >> ch;

    if (ch == 1) {
        int n;
        double c;
        cin >> n >> c;
        Wooden w(n, c);
        cout << fixed << setprecision(2) << w.calculateAmount() << endl;
    }
    else if (ch == 2) {
        double c;
        cin >> c;
        Electronics e(c);
        cout << fixed << setprecision(2) << e.calculateAmount() << endl;
    }
    else {
        cout << "Invalid choice." << endl;
    }

    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Easy_Pure Virtual Functions

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Imagine you are creating a hotel room pricing system with a class Hotel featuring a pure virtual function calculatePrice() to calculate prices based on nights and guests. Derived classes SingleRoom and DoubleRoom implement this function.

Base cost calculation:

For Single room - Rs. 100 per nightFor Double room - Rs. 150 per night

Additional cost calculation:

If guests stay for more than 5 nights, they avail a 10% discount. If guests stay for 5 or fewer nights, add Rs. 15 service charge.

The program takes user input for nights and guests and then calculates and displays costs for both room types.

Answer

```
// You are using GCC
#include <iostream>
using namespace std;

class Hotel {
public:
   virtual int calculatePrice(int n) = 0;
   virtual ~Hotel() {}
};

class SingleRoom : public Hotel {
public:
   int calculatePrice(int n) override {
      int cost = n * 100;
      return (n > 5) ? cost * 0.9 : cost + 15;
   }
};

class DoubleRoom : public Hotel {
public:
   int calculatePrice(int n) override {
      int cost = n * 150;
      return (n > 5) ? cost * 0.9 : cost + 15;
   }
};

int main() {
   int n, p;
   cin >> n >> p;

   SingleRoom s;
   DoubleRoom d;

   cout << "Single Room Cost: " << s.calculatePrice(n) << endl;
   cout << "Double Room Cost: " << d.calculatePrice(n) << endl;

   return 0;
}
```

Status : Correct *Marks : 10/10*

2. Problem Statement

Rohith is designing a program that consists of a base class Expression with a pure virtual function evaluate().

There are two derived classes PowerAB and PowerBA representing expressions of the form ab and ba respectively, where a and b are real numbers. The program takes the values of a and b as input and calculates and displays the power results for both types of expressions.

Note: Use the pow function from the math library to calculate the power value.

Answer

```
// You are using GCC
#include <iostream>
#include <cmath>
#include <iomanip>
using namespace std;

class Expression {
public:
    virtual double evaluate(double a, double b) = 0;
    virtual ~Expression() {}
};

class PowerAB : public Expression {
public:
    double evaluate(double a, double b) override {
        return pow(a, b);
    }
};

class PowerBA : public Expression {
public:
    double evaluate(double a, double b) override {
        return pow(b, a);
    }
};

int main() {
    double a, b;
    cin >> a >> b;

    PowerAB ab;
    PowerBA ba;

    cout << fixed << setprecision(2) << ab.evaluate(a, b) << endl;
    cout << fixed << setprecision(2) << ba.evaluate(a, b) << endl;

    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Easy_Abstract Classes

Attempt : 1
Total Mark : 20
Marks Obtained : 12.5

Section 1 : Coding

1. Problem Statement:

Write a Program to calculate the current bill.

Create a class currentBill with a virtual method double amount().

Create a Fan

Create a class Fan that extends currentBill with watts and hours as its public attributes and overrides the virtual function.

Create a class Light that extends currentBill with watts and hours as its public attributes and overrides the virtual function.

Create a class TV that extends currentBill with watts and hours as its public attributes and overrides the virtual function.

In the main method, prompt the user to enter the power rate of the appliance and the total hours used then create the necessary objects and call the methods.

Answer

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

// Abstract base class
class currentBill {
public:
   virtual double amount() = 0; // Pure virtual function
};

// Fan class inheriting from currentBill
class Fan : public currentBill {
public:
   int watts, hours;
   Fan(int w, int h) { watts = w; hours = h; } // Constructor to initialize values
   double amount() override {
      return (watts * hours * 1.0) / 1000; // Energy consumption in kWh
   }
};

// Light class inheriting from currentBill
class Light : public currentBill {
public:
   int watts, hours;
   Light(int w, int h) { watts = w; hours = h; } // Constructor to initialize values
   double amount() override {
      return (watts * hours * 1.0) / 1000; // Energy consumption in kWh
   }
};

// TV class inheriting from currentBill
class TV : public currentBill {
public:
   int watts, hours;
   TV(int w, int h) { watts = w; hours = h; } // Constructor to initialize values
   double amount() override {
      return (watts * hours * 1.0) / 1000; // Energy consumption in kWh
   }
};

int main() {
   // Variables to store input values for each appliance
   int fw, fh, lw, lh, tw, th;

   // Input: Watts and hours for Fan, Light, and TV
   cin >> fw >> fh;
   cin >> lw >> lh;
   cin >> tw >> th;

   // Create objects for each appliance
   Fan f(fw, fh);
   Light l(lw, lh);
   TV t(tw, th);

   // Calculate total energy consumption by summing up individual amounts
   double total = f.amount() + l.amount() + t.amount();
   total=(total)*1.5;
   // Output the total energy consumption formatted to 2 decimal places
   cout << fixed << setprecision(2) << total;

   return 0;
}
```

Status : Partially correct ***Marks : 2.5/10***

2. Problem Statement:

Create a base class named operationsBase with the following four virtual functions

void addition()

void subtraction()

void multiplication()

void division()

Create a derived class named operationsDerived that extends operationsBase with a and b as its private attributes and override the virtual functions.

Answer

```
#include <iostream>
class OperationsBase {
public:
   virtual void addition() = 0;
   virtual void subtraction() = 0;
   virtual void multiplication() = 0;
   virtual void division() = 0;
};

class OperationsDerived : public OperationsBase {
   int a, b;
public:
   OperationsDerived(int x, int y) : a(x), b(y) {}
   void addition() override { std::cout << (a + b) << " "; }
   void subtraction() override { std::cout << (a - b) << " "; }
   void multiplication() override { std::cout << (a * b) << " "; }
   void division() override { std::cout << (a / b) << " "; }
};

int main() {
   int x, y;
   std::cin >> x >> y;
   OperationsDerived obj(x, y);
   obj.addition();
   obj.subtraction();
   obj.multiplication();
   obj.division();
   return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Medium_Function Overloading

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are creating a shopping cart for an e-commerce website. Implement overloaded functions called calculateTotalPrice that can calculate the total price of different types of items, such as a single item, multiple items, or items with discounts. Each overloaded function should take the required parameters and return the calculated total price.

Calculate the total price for a single item without quantity or discounts.Calculate the total price for multiple items in quantity without any discounts.Calculate the total price for multiple items with quantity and a discount percentage.

Answer

```
#include <iostream>
```

```
#include <iomanip>
using namespace std;

double calculateTotalPrice(double p) {
   return p;
}

double calculateTotalPrice(double p, int q) {
   return p * q;
}

double calculateTotalPrice(double p, int q, double d) {
   return p * q * (1 - d / 100);
}

int main() {
   int c;
   while (cin >> c) {
      if (c == 1) {
         double p;
         cin >> p;
         cout << fixed << setprecision(2) << calculateTotalPrice(p) << "\n";
      } else if (c == 2) {
         double p;
         int q;
         cin >> p >> q;
         cout << fixed << setprecision(2) << calculateTotalPrice(p, q) << "\n";
      } else if (c == 3) {
         double p, d;
         int q;
         cin >> p >> q >> d;
         cout << fixed << setprecision(2) << calculateTotalPrice(p, q, d) << "\n";
      } else if (c == 4) {
         break;
      } else {
         cout << "Invalid choice\n";
      }
   }
   return 0;
}
```

Status : Correct   ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Medium_Operator Overloading

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Maria is developing a program to manage student records, with a specific focus on weight tracking. She has created a class named Student that represents student information and allows for the incrementing (++) of a student's weight by 1 kg through operator overloading with the friend function named operator.

Assist her in completing the program.

Answer

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

```
class Student {
   int id;
   double weight;
public:
   Student(int i, double w) : id(i), weight(w) {}
   friend Student operator++(Student &s, int) {
      Student temp = s;
      s.weight += 1;
      return temp;
   }
   void display() {
      cout << id << " " << fixed << setprecision(2) << weight << "\n";
   }
};

int main() {
   int id;
   double w;
   cin >> id >> w;
   Student s(id, w);
   s.display();
   s++;
   s.display();
   return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Medium_Virtual Functions

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Create a program that manages student information and grades for two types of students: Undergraduate (U) and Graduate (G). The program allows users to input a student's name, roll number, and type ('U' for Undergraduate or 'G' for Graduate). Depending on the type, the program calculates and displays the total grade for the student.

Classes and Virtual Functions: Student (Base Class)

Virtual Functions

virtual void inputGrades(): Handles grade input.

virtual void calculateGrade(): Calculates and displays the total grade.

UndergraduateStudent (Derived Class):

Inherits from Student.

Overrides virtual functions for grade input and grade calculation for Undergraduate students.

GraduateStudent (Derived Class):

Inherits from Student.

Overrides virtual functions for grade input and grade calculation for Graduate students.

Answer

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

class Student {
protected:
   string name, rollNumber;
public:
   Student(string n, string r) : name(n), rollNumber(r) {}
   virtual void inputGrades() = 0;
   virtual void calculateGrade() = 0;
   virtual ~Student() {}
};

class UndergraduateStudent : public Student {
   int midterm, finalExam;
public:
   UndergraduateStudent(string n, string r) : Student(n, r) {}
   void inputGrades() override {
      cin >> midterm >> finalExam;
   }
   void calculateGrade() override {
      cout << "Name: " << name << "\nRoll Number: " << rollNumber << "\nTotal
Grade: " << (midterm + finalExam) / 2 << "\n";
   }
};
```

```
class GraduateStudent : public Student {
  int research, presentation;
public:
  GraduateStudent(string n, string r) : Student(n, r) {}
  void inputGrades() override {
    cin >> research >> presentation;
  }
  void calculateGrade() override {
    cout << "Name: " << name << "\nRoll Number: " << rollNumber << "\nTotal Grade: " << (research + presentation) / 2 << "\n";
  }
};

int main() {
  string name, rollNumber;
  char type;
  cin >> name >> rollNumber >> type;
  Student *s;
  if (type == 'U') {
    s = new UndergraduateStudent(name, rollNumber);
  } else {
    s = new GraduateStudent(name, rollNumber);
  }
  s->inputGrades();
  s->calculateGrade();
  delete s;
  return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Medium_Pure Virtual Functions

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Design a program for a transport system operation. Create a base class Transport with a pure virtual function named operate. Implement two derived classes, TransportA and TransportB, which calculate and display the time taken to move between locations based on the transport type.

The program takes input for the number of transport systems and their operations, then outputs the time taken in seconds for each operation.

For TransportA: Time Taken = abs(start location - destination) * 2

For TransportB: Time Taken = abs(start location - destination) * 3

Answer

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

class Transport {
public:
   virtual void operate(int start, int dest) = 0;
   virtual ~Transport() {}
};

class TransportA : public Transport {
public:
   void operate(int start, int dest) override {
      cout << "Time taken: " << abs(start - dest) * 2 << " seconds\n";
   }
};

class TransportB : public Transport {
public:
   void operate(int start, int dest) override {
      cout << "Time taken: " << abs(start - dest) * 3 << " seconds\n";
   }
};

int main() {
   int n;
   cin >> n;
   while (n--) {
      char type;
      int start, dest;
      cin >> type >> start >> dest;
      Transport *t;
      if (type == 'P' || type == 'p') {
         t = new TransportA();
      } else {
         t = new TransportB();
      }
      t->operate(start, dest);
      delete t;
   }
```

```
    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Medium_Abstract Classes

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement:

Alphabetics Game:

You have to enter four letters for each uppercase letter you will get 10 marks and for each lowercase letter, you will get -5 marks.

Write a program to calculate the total score.

Create a base class with a virtual method void game(). Define this method in the Derived class and calculate the total score.

Answer

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

```
class AlphabeticsGame {
public:
   virtual void game() = 0;
   virtual ~AlphabeticsGame() {}
};

class ScoreCalculator : public AlphabeticsGame {
public:
   void game() override {
      char a, b, c, d;
      cin >> a >> b >> c >> d;
      int score = 0;
      for (char ch : {a, b, c, d}) {
         score += (isupper(ch) ? 10 : -5);
      }
      cout << "Score : " << score << "\n";
   }
};

int main() {
   ScoreCalculator s;
   s.game();
   return 0;
}
```

Status : Correct

Marks : 10/10

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Hard_Function Overloading

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

You are designing a traffic simulator. Implement overloaded functions called calculateSpeed that can calculate different speeds, such as the speed of a car, the speed of a train, and the speed of a plane. Each overloaded function should take the required inputs and return the calculated speed.

Formula:

speed = distance/time

speed = (distance/time) + acceleration*time

Example

Input:

100.5

10

5.5

6.6

Output:

10.05

65.05

76.05

Answer

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

double calculateSpeed(double d, double t) {
   return d / t;
}

double calculateSpeed(double d, double t, double a) {
   return (d / t) + (a * t);
}

int main() {
   double d, t, a1, a2;
   cin >> d >> t >> a1 >> a2;
   cout << fixed << setprecision(2) << calculateSpeed(d, t) << "\n";
   cout << fixed << setprecision(2) << calculateSpeed(d, t, a1) << "\n";
   cout << fixed << setprecision(2) << calculateSpeed(d, t, a2) << "\n";
   return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Hard_Operator Overloading

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Create a class Time to represent the departure time of a train, initialized with hours and minutes. Overload the pre-increment operator ++ to advance the departure time by one minute. Display the original and updated departure times.

Note: The time is represented in 24-hour format.

Answer

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

class Time {
    int h, m;
public:
    Time(int hours, int minutes) : h(hours), m(minutes) {}
    void display() { cout << h << "h " << m << "m\n"; }
    Time& operator++() {
        m++;
        if (m == 60) { h = (h + 1) % 24; m = 0; }
        return *this;
    }
};

int main() {
    int h, m;
    cin >> h >> m;
    Time t(h, m);
    t.display();
    ++t;
    t.display();
    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Hard_Virtual Functions

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Sharon is working on a population growth analysis program. She is studying how populations change over time.

She wants to write a program that calculates the growth rate of a population based on its initial and final sizes.

Help Sharon write a program that includes a base class Popul and a derived class Birth. Popul has a virtual function calcRate() to initialize the initial and final sizes. Class Birth inherits from Popul, overriding calcRate() to calculate the growth rate based on births.

Note: Growth Rate = (1000 * (final polulation - initial population) / initial population)

Answer

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

class Popul {
protected:
   int initial, final;
public:
   virtual void calcRate(int n, int p) {
      initial = n;
      final = p;
   }
};

class Birth : public Popul {
public:
   void calcRate(int n, int p) override {
      Popul::calcRate(n, p);
      double rate = (1000.0 * (final - initial)) / initial;
      cout << fixed << setprecision(2) << rate << endl;
   }
};

int main() {
   int n, p;
   cin >> n >> p;
   Birth b;
   b.calcRate(n, p);
   return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Hard_Pure Virtual Functions

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Write a program to calculate the total cost of a meal for a group of people.

The program has a base class, MenuItem, with an attribute price and a pure virtual function calculatePrice().

Two derived classes, Appetizer and MainCourse, inherit from MenuItem. They implement the pure virtual function to calculate the total cost based on the price of the item and the number of persons.

For Example,

If there are 4 people in the group, and if the number of appetizer items is 3 and the number of main course items is 4.

The appetizer prices are 44.25, 53.50, 75.99.The main course prices are 48.99, 53.25, 86.75, 92.00.The total cost is calculated by adding all the costs and multiplying the number of people.

For appetizers: (44.25 + 53.50 + 75.99) = 173.74For main courses: (48.99 + 53.25 + 86.75 + 92.00) = 280.99Total cost = (173.74 + 280.99) * 4 = 454.73 * 4 = 1818.92.

Answer

```
#include <iostream>
#include <iomanip>
#include <vector>
using namespace std;

class MenuItem {
protected:
   double price;
public:
   virtual double calculatePrice(int count) = 0;
};

class Appetizer : public MenuItem {
public:
   Appetizer(double p) { price = p; }
   double calculatePrice(int count) override {
      return price * count;
   }
};

class MainCourse : public MenuItem {
public:
   MainCourse(double p) { price = p; }
   double calculatePrice(int count) override {
      return price * count;
   }
};

int main() {
   int n, N, M;
   cin >> n >> N >> M;
```

```
    vector<Appetizer> appetizers;
    vector<MainCourse> mainCourses;
    double price, total = 0;

    for (int i = 0; i < N; ++i) {
        cin >> price;
        appetizers.emplace_back(price);
        total += appetizers.back().calculatePrice(1);
    }

    for (int i = 0; i < M; ++i) {
        cin >> price;
        mainCourses.emplace_back(price);
        total += mainCourses.back().calculatePrice(1);
    }

    total *= n;
    cout << fixed << setprecision(2) << "Rs. " << total << endl;
    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_COD_Hard_Abstract Classes

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : Coding

1. Problem Statement:

Create a base class ItemType with one virtual function

double calculateAmount()

Create a class wooden that extends ItemType with a number of items and cost as its private attributes. Obtain the data members and override the virtual function.

amount = number of items*cost

Create a class electronics that extend ItemType with cost as its private attribute. Obtain the data member and override the virtual function.

amount = 80% of the amount (20% discount)

In the main method, obtain a choice

If the choice is 1, create an object for the wooden class and call the method.

If the choice is 2, create an object for the electronics class and call the method.

Answer

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

class ItemType {
public:
   virtual double calculateAmount() = 0;
};

class Wooden : public ItemType {
private:
   int numItems;
   double cost;
public:
   Wooden(int n, double c) : numItems(n), cost(c) {}
   double calculateAmount() override {
      return numItems * cost;
   }
};

class Electronics : public ItemType {
private:
   double cost;
public:
   Electronics(double c) : cost(c) {}
   double calculateAmount() override {
      return cost * 0.8;
   }
};

int main() {
   int choice;
```

```
    cin >> choice;
    double amount;

    if (choice == 1) {
        int n;
        double cost;
        cin >> n >> cost;
        Wooden item(n, cost);
        amount = item.calculateAmount();
    } else if (choice == 2) {
        double cost;
        cin >> cost;
        Electronics item(cost);
        amount = item.calculateAmount();
    }

    cout << fixed << setprecision(2) << amount << endl;
    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_CY_Function Overloading

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Ragul is the head of the UPSC (Union Public Service Commission) and he needs to schedule interviews for candidates applying for various government positions. The interview slots are divided into three available time slots: Slot A, Slot B, and Slot C. Additionally, each candidate can provide a preferred slot.

Implement a program that takes candidate details and schedules their interviews based on the given constraints.

Create a class called UPSCInterviewScheduler that will manage the interview scheduling process. This class will have the following methods and functionalities:

In the UPSCInterviewScheduler class, implement function overloading for the printSlot method. Create two versions of the method:

printSlot(const string&amp; slot, const string&amp; preferredSlot): This version of the method will be used to print the assigned slots as described above.

printSlot(const string&amp; slot): This version of the method will be used to print a regular slot without considering the preferred slot.

Answer

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

class UPSCInterviewScheduler {
public:
    void printSlot(const string& slot, const string& preferredSlot) {
        if (slot == preferredSlot) {
            cout << "- " << slot << " (Preferred)" << endl;
        } else {
            cout << "- " << slot << endl;
        }
    }

    void printSlot(const string& slot) {
        cout << "- " << slot << endl;
    }
};

int main() {
    UPSCInterviewScheduler scheduler;
    vector<vector<string>> candidates;
    vector<string> preferredSlots;
    string slot1, slot2, slot3, preferred;
    int candidateCount = 0;

    while (cin >> slot1 >> slot2 >> slot3 >> preferred) {
        candidates.push_back({slot1, slot2, slot3});
        preferredSlots.push_back(preferred);
        candidateCount++;
    }

    for (int i = 0; i < candidateCount; i++) {
        cout << "Candidate " << i + 1 << ":" << endl;
        scheduler.printSlot(candidates[i][0], preferredSlots[i]);
        scheduler.printSlot(candidates[i][1], preferredSlots[i]);
        scheduler.printSlot(candidates[i][2], preferredSlots[i]);
        cout << endl;
    }

    return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Create a function 'add'. The function add should perform the following operation.

The first argument should be the data type of the remaining arguments (str or int) (inp_type)It should be able to get any number of arguments (inp1, inp2, inp3,....)if the data type is 'str', it should perform string concatenation.if the data type is 'int', it should perform an addition.

In the main program, get the user input as below:

inp_type -> type of input

inp1 -> arg1

inp2 -> arg2

inp3 -> arg3

Execute the below statements:

add(inp_type,inp1)

add(inp_type,inp1,inp2)

add(inp_type, inp1,inp2,inp3)

Note: Use the same naming convention as mentioned above.

Answer

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

template <typename T>
void add(T first) {
    cout << first << endl;
}

template <typename T>
void add(T first, T second) {
    cout << first + second << endl;
}

template <typename T>
void add(T first, T second, T third) {
    cout << first + second + third << endl;
}

int main() {
    string inp_type;
    cin >> inp_type;

    if (inp_type == "int") {
        int a, b, c;
        cin >> a >> b >> c;
        add(a);
        add(a, b);
        add(a, b, c);
    } else if (inp_type == "str") {
        string a, b, c;
        cin >> a >> b >> c;
        add(a);
        add(a, b);
        add(a, b, c);
    }

    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_CY_Operator Overloading

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Implement a Matrix class that represents a two-dimensional matrix of integers. Overload the unary - operator to negate all elements of the matrix and return a new matrix with the negated values.

Create a class called Matrix and Matrix operator-() overloads the unary - operator to negate all elements of the matrix. It returns a new Matrix object with the negated values.

Answer

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

```
class Matrix {
   int rows, cols;
   vector<vector<int>> mat;
public:
   Matrix(int r, int c) : rows(r), cols(c), mat(r, vector<int>(c)) {}
   void input() {
      for (int i = 0; i < rows; ++i)
         for (int j = 0; j < cols; ++j)
            cin >> mat[i][j];
   }
   void print() const {
      for (const auto& row : mat) {
         for (int val : row) cout << val << " ";
         cout << "\n";
      }
   }
   Matrix operator-() const {
      Matrix neg(rows, cols);
      for (int i = 0; i < rows; ++i)
         for (int j = 0; j < cols; ++j)
            neg.mat[i][j] = -mat[i][j];
      return neg;
   }
};

int main() {
   int r, c;
   cin >> r >> c;
   Matrix mat(r, c);
   mat.input();
   cout << "Original Matrix:\n";
   mat.print();
   cout << "Negated Matrix:\n";
   (-mat).print();
   return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Write a C++ program that defines a Matrix class capable of performing various matrix operations. The class should allow matrix multiplication, transpose, and input/output operations.

In the main function, the program should take two matrices as input, calculate their product (if valid), and display the result. Additionally, it should compute and display the transpose of the first matrix.

Answer

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

class Matrix {
   vector<vector<int>> mat;
   int rows, cols;
public:
   Matrix(int r, int c) : rows(r), cols(c), mat(r, vector<int>(c)) {}
   void input() {
      for (int i = 0; i < rows; ++i)
         for (int j = 0; j < cols; ++j)
            cin >> mat[i][j];
   }
   void print() const {
      for (const auto& row : mat) {
         for (int val : row) cout << val << " ";
         cout << "\n";
      }
   }
   Matrix transpose() const {
      Matrix trans(cols, rows);
      for (int i = 0; i < rows; ++i)
         for (int j = 0; j < cols; ++j)
            trans.mat[j][i] = mat[i][j];
      return trans;
   }
   Matrix operator*(const Matrix& other) const {
      Matrix prod(rows, other.cols);
      for (int i = 0; i < rows; ++i)
         for (int j = 0; j < other.cols; ++j)
```

```
                for (int k = 0; k < cols; ++k)
                    prod.mat[i][j] += mat[i][k] * other.mat[k][j];
        return prod;
    }
};

int main() {
    int r1, c1, r2, c2;
    cin >> r1 >> c1;
    Matrix m1(r1, c1);
    m1.input();
    cin >> r2 >> c2;
    Matrix m2(r2, c2);
    m2.input();

    if (c1 == r2) {
        cout << "Matrix product:\n";
        (m1 * m2).print();
    }
    cout << "Transpose of the first matrix:\n";
    m1.transpose().print();
    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_CY_Virtual Functions

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Prabha needs to create a program for a ticket reservation system.

Assist her in creating a program that uses the base class Reservation with the virtual function calculate(), which will be overridden in the derived classes VIP and Standard.

VIP reservations receive age-based discounts:

21% for ages 4–12.15% for ages 13–25.10% for ages 26–64.

Standard reservations receive age-based discounts:

18% for ages 4–12.14% for ages 13–25.5% for ages 26–64.

Write a program that takes input for VIP and standard ticket prices, along with the number of each reservation and the age of the customers. Calculate and display the total prices for both VIP and standard reservations.

Answer

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

class Reservation {
public:
   virtual double calculate(double price, int age, int count) = 0;
};

class VIP : public Reservation {
public:
   double calculate(double price, int age, int count) override {
      double discount = (age >= 4 && age <= 12) ? 0.21 :
                        (age >= 13 && age <= 25) ? 0.15 :
                        (age >= 26 && age <= 64) ? 0.10 : 0;
      return count * price * (1 - discount);
   }
};

class Standard : public Reservation {
public:
   double calculate(double price, int age, int count) override {
      double discount = (age >= 4 && age <= 12) ? 0.18 :
                        (age >= 13 && age <= 25) ? 0.14 :
                        (age >= 26 && age <= 64) ? 0.05 : 0;
      return count * price * (1 - discount);
   }
};

int main() {
   double vipPrice, stdPrice;
   int vipAge, stdAge, vipCount, stdCount;

   cin >> vipPrice >> vipAge >> vipCount;
   cin >> stdPrice >> stdAge >> stdCount;
   VIP vip;
   Standard std;

   cout << fixed << setprecision(2);
   cout << "VIP ticket price: Rs." << vip.calculate(vipPrice, vipAge, vipCount) <<
endl;
   cout << "Standard ticket price: Rs." << std.calculate(stdPrice, stdAge, stdCount)
<< endl;

   return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Arun is developing a time conversion program that consists of a base class called Time and two derived classes named TimeToSeconds and TimeToMinutes.

class Time - stores hours and minutes and has a virtual function display() that prints the input time.class TimeToSeconds - overrides the display method to convert the time to seconds.class TimeToMinutes - overrides the display method to convert the time to minutes.

To assist Arun, we need to create a program that accepts input for hours and minutes, creates instances of the derived classes, and displays the original time, the time in seconds, and the time in minutes.

Answer

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

class Time {
protected:
   int h, m;
public:
   Time(int hr, int min) : h(hr), m(min) {}
   virtual void display() {
        cout << "Time: " << h << "h " << m << "m\n";
    }
};

class TimeToSeconds : public Time {
public:
    TimeToSeconds(int hr, int min) : Time(hr, min) {}
    void display() override {
        cout << (h * 3600 + m * 60) << " seconds\n";
    }
};

class TimeToMinutes : public Time {
public:
    TimeToMinutes(int hr, int min) : Time(hr, min) {}
    void display() override {
        cout << (h * 60 + m) << " minutes\n";
    }
};

int main() {
    int hr, min;
    cin >> hr >> min;

    Time t(hr, min);
    TimeToSeconds ts(hr, min);
    TimeToMinutes tm(hr, min);

    t.display();
    ts.display();
    tm.display();

    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_CY_Pure Virtual Functions

Attempt : 1
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Problem Statement

Sharon wants to design a program to analyze experimental results.

Create a base class, Experiment, with a pure virtual function analyzeResults(). Implement two derived classes, ChemistryExperiment and PhysicsExperiment, to analyze chemistry and physics experiments, respectively.

For the chemistry experiment, calculate the value: Product = Reactant A * Reactant B

For the physics experiment, calculate the value: Final Velocity = Initial Velocity + (Acceleration * Time)

Answer

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

class Experiment {
public:
    virtual void analyzeResults() = 0;
    virtual ~Experiment() {}
};

class ChemistryExperiment : public Experiment {
    double a, b;
public:
    ChemistryExperiment(double reactantA, double reactantB) : a(reactantA),
b(reactantB) {}
    void analyzeResults() override {
        cout << fixed << setprecision(2) << a * b << endl;
    }
};

class PhysicsExperiment : public Experiment {
    double u, a, t;
public:
    PhysicsExperiment(double initialVelocity, double acceleration, double time) :
u(initialVelocity), a(acceleration), t(time) {}
    void analyzeResults() override {
        cout << fixed << setprecision(2) << (u + a * t) << " m/s" << endl;
    }
};

int main() {
    char expType;
    cin >> expType;

    if (expType == 'C' || expType == 'c') {
        double a, b;
        cin >> a >> b;
        ChemistryExperiment chem(a, b);
        chem.analyzeResults();
    } else if (expType == 'P' || expType == 'p') {
        double u, a, t;
        cin >> u >> a >> t;
        PhysicsExperiment phys(u, a, t);
        phys.analyzeResults();
    }

    return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem statement

You are tasked with developing a simple inventory management system for a retail shop called "Dukaan." The shop sells various items, each with a unique item ID and an associated price. Your goal is to create a program that allows the shopkeeper to set prices for items and then display the prices of all the items in the shop.

To implement this, you have been provided with a base class named Shop, which includes a pure virtual function setPrice to set the price of an item and a function displayPrice to display the prices of all the items in the shop.

Your task is to create a derived class named ConcreteShop that inherits from the Shop class. The ConcreteShop class should override the setPrice function to store the item IDs and their corresponding prices in two arrays: itemId and itemPrice.

Answer

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

class Shop {
public:
    virtual void setPrice(int id, int price) = 0;
    virtual void displayPrice() = 0;
};

class ConcreteShop : public Shop {
    int itemId[10], itemPrice[10], n;
public:
    ConcreteShop(int size) : n(size) {}
    void setPrice(int id, int price) {
      static int i = 0;
      itemId[i] = id;
      itemPrice[i] = price;
      i++;
    }
    void displayPrice() {
      for (int i = 0; i < n; i++)
        cout << "The Price of item with Id " << itemId[i] << " is " << itemPrice[i] <<
"\n";
    }
};

int main() {
    int n, id, price;
    cin >> n;
    ConcreteShop shop(n);
    for (int i = 0; i < n; i++) {
      cin >> id >> price;
      shop.setPrice(id, price);
    }
    shop.displayPrice();
    return 0;
}
```

Status : Correct ***Marks : 10/10***

VIT - Vellore

Name: SOUMYO JYOTI SAHA .
Email: soumyojyoti.saha2021@vitstudent.ac.in
Roll no: 21BCE4007
Phone: 9898989898
Branch: VINILA JINNY_OOPS
Department: admin
Batch: VL2024250501961
Degree: admin

Scan to verify results

BCSE102P_Structured and Object Oriented Programming Lab_VL2024250501961

VIT V_Structured and OOP_Lab 7_CY_Abstract Classes

Attempt : 2
Total Mark : 20
Marks Obtained : 20

Section 1 : Coding

1. Create a base class Calculations with Three virtual function

double amount()

double Time()

void display()

Create a class Scooty that extends Calculations with distance as its public attribute and overrides the virtual functions as follows.

double amount() - it will return the total amount after 5% discount (Price to travel 1 km is Rs.100, Speed of the Scooty is 40 km/hr)

double Time() - it will return the total hours taken to travel.

Void display() - it will display the Amount and time taken.

Create a class Bike that extends Calculations with distance as its public attribute and overrides the virtual functions as follows.

double amount() - it will return the total amount after 10% discount (Price to travel 1 km is Rs.150 and Speed of the bike is 60Km/hr)

double Time() - it will return the total hours taken to travel.

Void display() - it will display the Amount and time taken.

Create a class Car that extends Calculations with distance as its public attribute and overrides the virtual functions as follows.

double amount() - it will return the total amount after 20% discount (Price to travel 1 km is Rs.200 and Speed of the car is 80Km/hr)

double Time() - it will return the total hours taken to travel.

Void display() - it will display the Amount and time taken.

In the main method, prompt the user to enter the distance and create the necessary objects and call the methods.

Answer

```
// You are using GCC
#include <iostream>
using namespace std;

class Calculations {
public:
   virtual double amount() = 0;
   virtual double Time() = 0;
   virtual void display() = 0;
   virtual ~Calculations() {}
};

class Scooty : public Calculations {
public:
   double distance;

   Scooty(double d) : distance(d) {}
```

```
    double amount() override {
        double total = distance * 100;
        return total * 0.95; // 5% discount
    }

    double Time() override {
        return distance / 40.0;
    }

    void display() override {
        cout << "Scooty" << endl;
        cout << "Time Taken(in hours) : " << Time() << endl;
        cout << "Amount : " << amount() << endl;
    }
};

class Bike : public Calculations {
public:
    double distance;

    Bike(double d) : distance(d) {}

    double amount() override {
        double total = distance * 150;
        return total * 0.90; // 10% discount
    }

    double Time() override {
        return distance / 60.0;
    }

    void display() override {
        cout << "Bike" << endl;
        cout << "Time Taken(in hours) : " << Time() << endl;
        cout << "Amount : " << amount() << endl;
    }
};

class Car : public Calculations {
public:
    double distance;
    Car(double d) : distance(d) {}

    double amount() override {
        double total = distance * 200;
        return total * 0.80; // 20% discount
    }

    double Time() override {
        return distance / 80.0;
    }

    void display() override {
        cout << "Car" << endl;
        cout << "Time Taken(in hours) : " << Time() << endl;
        cout << "Amount : " << amount() << endl;
    }
};

int main() {
    double distance;
    cin >> distance;

    Scooty scooty(distance);
    Bike bike(distance);
    Car car(distance);

    scooty.display();
    bike.display();
    car.display();

    return 0;
}
```

Status : Correct ***Marks : 10/10***

2. Problem Statement

Create a base class Account with 3 virtual functions

void display()

void deposit(double amount)

void withdrawl(double amount)

Create a class savingsAccount that extends Account with accno, balance, and name as its private variables and override the virtual methods.

Create a class currentAccount that extends Account with accno, balance, name, draftlimit, and city as its private variables and override the virtual methods.

In the main method, create the objects and call the methods

Answer

```
#include<bits/stdc++.h>
#include<string>
using namespace std;
class Account
{
public:
  virtual void display() = 0;
  virtual void deposit(double amount) = 0;
  virtual void withdrawl(double amount) = 0;
};


class savingsAccount: public Account {
    int a;
    string n;
    double b;
public:
    savingsAccount(int x, string y, double z) { a = x; n = y; b = z; }
    void display() { cout << a << " " << n << " " << fixed << setprecision(2) << b <<
endl; }
    void deposit(double x) { b += x; cout << "Balance after deposit:" << fixed <<
setprecision(2) << b << endl; }
    void withdrawl(double x) { b -= x; cout << "Balance after withdrawl:" << fixed <<
setprecision(2) << b << endl; }
};

class currentAccount: public Account {
    int a, d;
    string n, c;
    double b;
public:
  currentAccount(int x, string y, double z, int m, string s) { a = x; n = y; b = z; d =
m; c = s; }
  void display() { cout << a << " " << n << " " << fixed << setprecision(2) << b << " "
<< d << " " << c << endl; }
  void deposit(double x) { b += x; cout << "Balance after deposit:" << fixed <<
setprecision(2) << b << endl; }
  void withdrawl(double x) { b -= x; cout << "Balance after withdrawl:" << fixed <<
setprecision(2) << b << endl; }
};

int main() {
  int ch;
  cin >> ch;

  if (ch == 1) {
    int a;
    string n;
    double b;
    cin >> a >> n >> b;
    savingsAccount s(a, n, b);
    s.display();
    int op;
    double amt;
    cin >> op >> amt;
    op == 1 ? s.deposit(amt) : s.withdrawl(amt);
    s.display();
  } else {
    int a, d;
    string n, c;
    double b;
    cin >> a >> n >> b >> d >> c;
    currentAccount k(a, n, b, d, c);
    k.display();
    int op;
    double amt;
    cin >> op >> amt;
    op == 1 ? k.deposit(amt) : k.withdrawl(amt);
    k.display();
  }

  return 0 ;
}
```

Status : Correct *Marks : 10/10*