**TUT-6**

**NAME-SANCHIT JAIN**

**BATCH – B-7**

**ENROLL – 21103192**

Output.

In Derived

In Derived

Q.2  
Output. In Derived

In Base

Q.3

There is compiler error in line “Base b;”

Q.4

Compiler Error: Derived is abstract.

Q.5

No. Constructor Cannot be virtual.

Q.6

Output. a>b

Q.7

Constructor: Base

Constructor: Derived

Destructor: Derived

Destructor: Base

Q.8

Output. C:fun()

Q.9

Output. In Base

#include <iostream>

using namespace std;

class shape

{

public:

    virtual float area() = 0;

    virtual float perimeter() = 0;

    virtual ~shape()

    {

    }

};

class rectangle : public shape

{

    int l;

    int b;

public:

    rectangle(int a = 0, int z = 0)

    {

        l = a;

        z = b;

    }

    float area()

    {

        return l \* b;

    }

    float perimeter()

    {

        return 2 \* (l \* b);

    }

};

class square : public shape

{

    int l;

public:

    square(int l = 0)

    {

        this->l = l;

    }

    float area()

    {

        return l \* l;

    }

    float perimeter()

    {

        return 4 \* l;

    }

};

class circle : public shape

{

    int r;

public:

    circle(int r = 0)

    {

        this->r = r;

    }

    float area()

    {

        return 3.14 \* r \* r;

    }

    float perimeter()

    {

        return 2 \* 3.14 \* r;

    }

};

int main()

{

    shape \*ptr[6];

    ptr[0] = new rectangle(8, 9);

    cout << "Area of rect: " << ptr[0]->area() << endl;

    cout << "Perimeter of rect: " << ptr[0]->perimeter() << endl;

    ptr[1] = new square(5);

    cout << "Area of square: " << ptr[1]->area() << endl;

    cout << "Perimeter of square: " << ptr[1]->perimeter() << endl;

    ptr[2] = new circle(10);

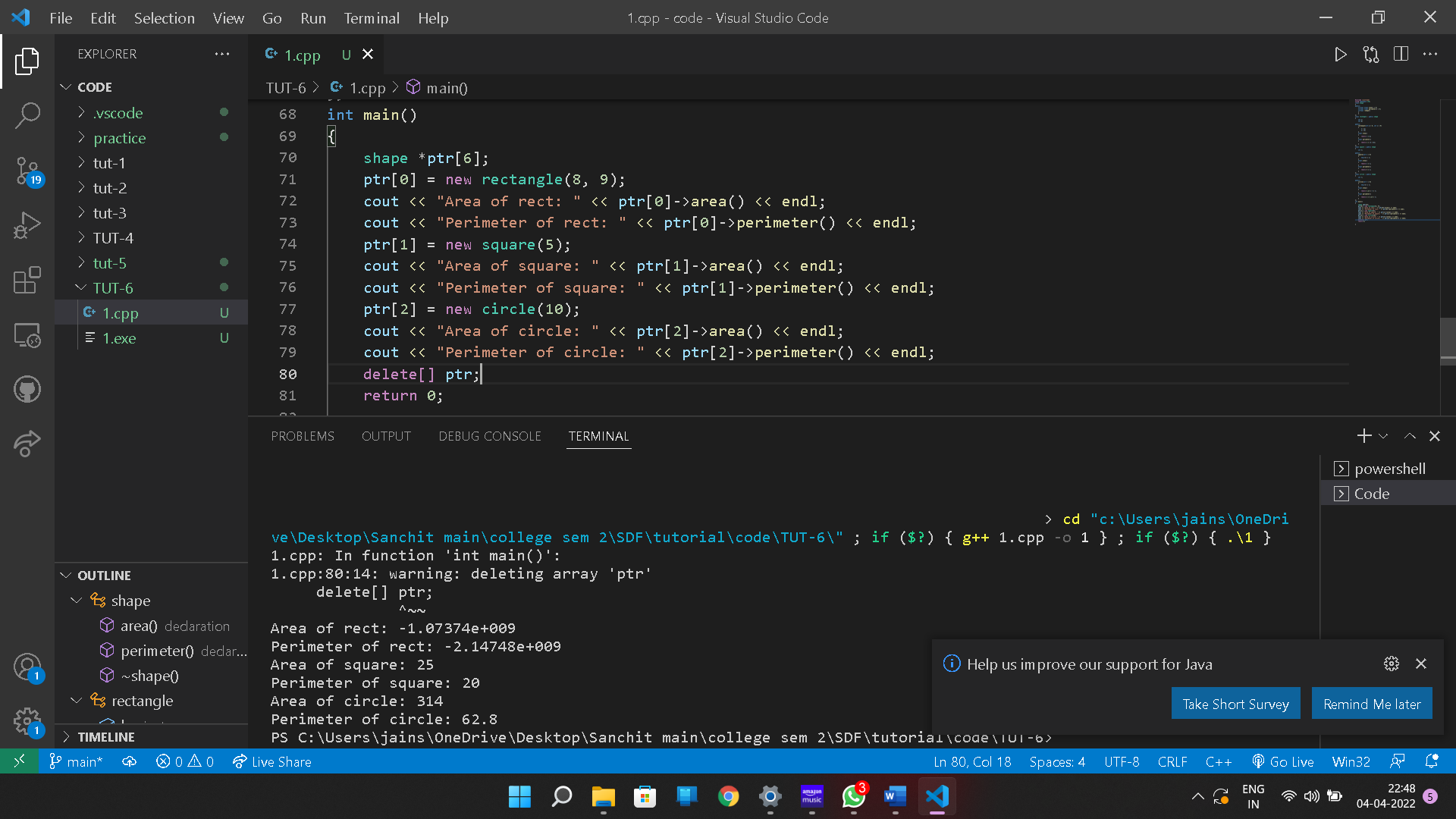
    cout << "Area of circle: " << ptr[2]->area() << endl;

    cout << "Perimeter of circle: " << ptr[2]->perimeter() << endl;

    delete[] ptr;

    return 0;

}



#include <iostream>

using namespace std;

class person

{

    static int flag;

    string name;

    int age;

public:

    virtual void getdata()

    {

        if (flag == 1)

        {

            cout << "Enter name: ";

            getline(cin, name);

            flag = 0;

        }

        else

        {

            cout << "Enter name: ";

            cin.ignore(1, '\n');

            getline(cin, name);

        }

        cout << "Enter age: ";

        cin >> age;

    }

    virtual void putdata()

    {

        cout << "Name: " << name << "\nAge: " << age << endl;

    }

    virtual ~person()

    {

    }

};

int person::flag = 1;

class professor : public person

{

    int publications, cur;

    static int cur\_id;

public:

    professor()

    {

        cur = cur\_id++;

    }

    void getdata()

    {

        person::getdata();

        cout << "Enter no. of publication: ";

        cin >> publications;

    }

    void putdata()

    {

        person::putdata();

        cout << "Publications: " << publications << "\nCur\_id: " << cur << endl;

    }

};

int professor::cur\_id = 1;

class student : public person

{

    int marks[6], cur;

    int sum = 0;

    static int cur\_id;

public:

    student()

    {

        cur = cur\_id++;

    }

    void getdata()

    {

        person::getdata();

        cout << "Enter marks of 6 subjects: ";

        for (int i = 0; i < 6; i++)

        {

            cin >> marks[i];

            sum += marks[i];

        }

    }

    void putdata()

    {

        person::putdata();

        cout << "Sum of marks: " << sum << "\nCur\_id: " << cur << endl;

    }

};

int student::cur\_id = 1;

int main()

{

    person \*ptr;

    person \*ptr1;

    ptr = new professor;

    ptr->getdata();

    ptr->putdata();

    ptr1 = new student;

    ptr1->getdata();

    ptr1->putdata();

    delete ptr;

    delete ptr1;

    return 0;

}

