

Quiz 03

Due May 17 at 11:59pm

Points 5

Questions 5

Available until May 17 at 11:59pm

Time Limit 20 Minutes

Instructions

Answer the following questions.

This quiz is no longer available as the course has been concluded.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	8 minutes	5 out of 5

Score for this quiz: **5** out of 5

Submitted May 17 at 11:25am

This attempt took 8 minutes.

Question 1	1 / 1 pts

Consider the class **Person**, consisting of data components **name** and **nationality**, defined as follows:

```
class Person {  
    public:  
        void setName(string name);  
        string getName() const;  
        void setNationality(string nationality);  
        string getNationality() const;  
    private:  
        string name;  
        string nationality;  
};
```

Class **Student** inherits from **Person**, with an additional data component **field**:

```
class Student : public Person {  
    public:  
        void setField(string field);  
        string getField() const;  
        void printInfo() const;  
    private:  
        string field;  
};
```

Assume that **prsn** is an object of class **Person**, and **stdnt** is an object of class **Student**, defined in **main()**:

Person prsn;

Student stdnt;

Which one of the following statements raises an error?

☐ `stdnt.setName("Alex");`

Correct!

- ☐ prsn.setNationality("American");
- ☐ stdnt.getName();
- ☒ prsn.setField("CS");

Objects of class Person do not include member function setField().

Question 2**1 / 1 pts**

Consider the class definitions **Person** and **Student**, given in the previous question. In the definition of **printInfo()** function:

```
void Student::printInfo() const {  
  
}
```

which one of the following statements raises an error?

- ☐ cout<<field<<endl;
- ☒ cout<<name<<endl;

name is a private field in class Person. Derived class Student does not have access to it directly. It must instead use getName() function.

- ☐ cout<<getNationality()<<endl;
- ☐ cout<<getName()<<endl;

Correct!

Question 3**1 / 1 pts**

Let's modify class **Person** as follows, where its data components are **protected** rather than **private**.

```
class Person {  
    public:  
        void setName(string name);  
        string getName() const;  
        void setNationality(string nationality);  
        string getNationality() const;  
    protected:  
        string name;  
        string nationality;  
};
```

Let's keep class **Student** as before:

```
class Student : public Person {  
    public:  
        void setField(string field);  
        string getField() const;  
        void printInfo() const;  
    private:  
        string field;  
};
```

Let's assume that function **printInfo()** and **main()** are defined as follows:

```
void Student::printInfo() const {  
    cout<<name<<endl;  
}
```

```
int main() {
```

```
int main() {  
    Student student;  
    cout<<student.name<<endl;  
    return 0;  
}
```

Which one of the following cases is correct?

- ☐ both printInfo() and main() cause errors.
- ☒ main() causes an error, but printInfo() does not.

printInfo() does not cause an error. Access to name is possible in Student class directly, since it is defined as protected in Person.

main() causes an error. name is not accessible outside class Person (except for the derived classes, i.e., Student), since it is defined as protected.

- ☐ Neither printInfo() nor main() cause errors.
- ☐ printInfo() causes an error, but main() does not.

Correct!

Question 4

1 / 1 pts

Consider the class definitions **Person** and **Student**, given in the previous question.

Let **setName()** be defined in **Person** as follows:

```
void Person::setName(string name) {  
    this -> name = name;  
}
```

Let this function be overridden in the derived class **Student** as follows:

```
void Student::setName(string name) {  
    this -> name = "Joe";  
}
```

Let **main()** be defined as:

```
int main() {  
    Student stdnt;  
    stdnt.setName("Ashley");  
    cout<<stdnt.getName()<<endl;  
  
    return 0;  
}
```

What is the output of this program?

Correct!

☒ Joe

Overridden setName() is invoked.

☐ Ashley

Question 5**1 / 1 pts**

Consider the class definitions **Person** and **Student**, given in the previous questions.

Let **setName()** be defined in **Person** as follows:

```
void Person::setName(string name) {  
    this -> name = name;  
}
```

Let this function be overridden in the derived class **Student**, where it simply calls **setName()** from base class.

```
void Student::setName(string name) {  
  
}
```

Which one is the correct way to fill out the missing part?

- ☐ setName(name);
- ☐ Student::setName(name);
- ☒ Person::setName(name);

Correct!**Quiz Score: 5 out of 5**