# Quiz 04

**Due** May 18 at 11:59pm **Points** 13 **Questions** 13 **Available** until May 18 at 11:59pm **Time Limit** 30 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	24 minutes	13 out of 13

Score for this quiz: **13** out of 13 Submitted May 18 at 3:38pm This attempt took 24 minutes.

Question 1	1 / 1 pts

```
Consider the definition of class Person as follows:
class Person {
  public:
    void setName(string name);
    string getName() const;
    void printlnfo() const;
  protected:
    string name;
};
void Person::setName(string name) {
  this -> name = name;
}
string Person::getName() const {
  return name;
void Person::printlnfo() const {
  cout<<"Name: "<<getName()<<endl;
}
In function main(), object person1 is defined as follows:
Person person1;
person1.setName("Bob");
At this point, how can you create a pointer perPtr to person1?
   Person perPtr = &person1;
```

### Correct!

Person \*perPtr = &person1;

	" is the type of poi of person1.	nters to Pers	son objects. "&" ge	ets the
<ul><li>Person</li></ul>	*perPtr = person1;			
Person	perPtr = person1;			

	Question 2	1 / 1 pts
	In the previous question, after defining <b>perPtr</b> as the pointer t <b>person1</b> , how can you call <b>printInfo()</b> that <b>perPtr</b> point to?	0
	operPtr.printInfo();	
	oprintInfo();	
orrect!	perPtr -> printInfo();	
	<pre>person1 -&gt; printlnfo();</pre>	

Question 3 1/1 pts

In addition to the definitions in the Question 1, let class Student be defined as follows:

```
class Student : public Person {
  public:
     void setField(string field);
     string getField() const;
     void printlnfo() const;
  private:
     string field;
};
void Student::setField(string f) {
  field = f;
}
string Student::getField() const {
  return field:
}
void Student::printlnfo() const {
  Person::printlnfo();
  cout<<"Field: "<<getField()<<endl;
}
In function main(), let perPtr point to a Student object:
Student person2;
person2.setName("Ashley");
person2.setField("CS");
Person *perPtr = &person2;
```

What would be the result of invoking **printlnfo()** that **perPtr** points to?

							,
r		DO:	DA	$\wedge$	0	ð.	
u	u			œ	اصا	a.	

Correct!

Name: Ashley

Since perPtr is a pointer to Person objects, printlnfo() defined in Person class is invoked.

Name: Ashley
Field: CS

## **Question 4**

1 / 1 pts

The previous question is called static polymorphism. How can we modify **printlnfo()** in **Person** class to make polymorphism dynamic?

- void printlnfo() const override;
- virtual void printlnfo() const;
- dynamic void printlnfo() const;

# **Question 5**

1 / 1 pts

	After making the polymorphism dynamic, what would be the result of invoking <b>printInfo()</b> that <b>perPtr</b> points to, considering the following definitions in <b>main()</b> :
	Student person2; person2.setName("Ashley"); person2.setField("CS");
	Person *perPtr = &person2
	No answer text provided.
Correct!	Name: Ashley  Field: CS
	Name: Ashley

	Question 6	1 / 1 pts
	A static data member of a class can be different for different of that class	bjects of
	O True	
Correct!	False	

Question 7 1 / 1 pts

<ul><li>Circle::circleNums = 5;</li><li>circleNums = 5;</li></ul>
circleNums = 5;
Question 8 1 / 1 pts
Static data member of a class needs to be initialized outside the class.
True
O False
Question 9 1 / 1 pts
Static member function of a class can access non-static data members of that class.
O True
<ul><li>False</li></ul>

	Question 10	1 / 1 pts
	Pointer <b>this</b> is available in a static member function.	
	O True	
Correct!	False	

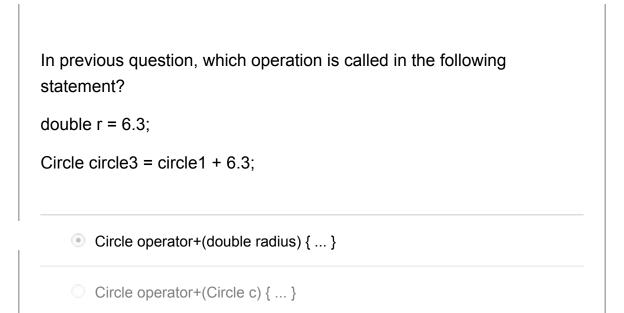
Question 11	1 / 1 pts

```
Consider class Circle
class Circle {
  public:
     Circle operator+(Circle c) { ... }
     Circle operator+(double radius) { ... }
  private:
     double radius;
};
with overloaded addition operator:
Circle operator+(Circle c) { ... }
and
Circle operator+(double radius) { ... }
Let circle1 and circle2 be two Circle objects in main(). Which operator
is called in the following statement?
Circle circle3 = circle1 + circle2;
    Circle operator+(double radius) { ... }
    Circle operator+(Circle c) { ... }
```

Question 12 1 / 1 pts

Correct!

Correct!



# In previous question, which operation is called in the following statement? double r = 6.3; Circle circle3 = 6.3 + circle1; Circle operator+(Circle c) { ... } This statement does not compile successfully. Circle operator+(double radius) { ... }

Quiz Score: 13 out of 13