When taking the question, it would appear as:

What is polymorphism? (in reference to objected oriented programming)

- The process of obtaining reusable methods and data fields from a parent class.
- The process of creating a variable that can take the form of any one of its subclasses.
- The ability to dynamically retype a variable to another type within the programming language at runtime.
- The process of structuring data types and functions into a higher level component.
- The process of hiding the concrete implementation from the clients. That way the client can focus on what is being processed rather than how it is being processed.

When grading the question, it would appear as:

What is polymorphism? (in reference to objected oriented programming)

  - The process of obtaining reusable methods and data fields from a parent class.

    - The process of creating a variable that can take the form of any one of its subclasses.

      - The ability to dynamically retype a variable to another type within the programming language at runtime.

        - The process of structuring data types and functions into a higher level component.

          - The process of hiding the concrete implementation from the clients. That way the client can focus on what is being processed rather than how it is being processed.

Comment: (given as feedback)

- No comment specified.

Hint:

- No hint specified.
When taking the question, it would appear as:

What is an abstract class?

- It is what C++ and some other language use to accomplish the same tasks as Java interfaces.

- It is a class that contains one or more abstract methods and defines a kind of class. Therefore, it is a model for subclasses and can't be instantiated directly.

- A class that is outlined in pseudo code and can be written in a programming language of your choice.

- It is a class that contains only static methods and variables, so that an instance of the class never needs to be created.

- The process of hiding the concrete implementation from the clients. That way the client can focus on what is being processed rather than how it is being processed.

When grading the question, it would appear as:

What is an abstract class?

It is what C++ and some other language use to accomplish the same tasks as Java interfaces.

It is a class that contains one or more abstract methods and defines a kind of class. Therefore, it is a model for subclasses and can't be instantiated directly.

A class that is outlined in pseudo code and can be written in a programming language of your choice.

It is a class that contains only static methods and variables, so that an instance of the class never needs to be created.

The process of hiding the concrete implementation from the clients. That way the client can focus on what is being processed rather than how it is being processed.

Comment: (given as feedback)

- No comment specified.

Hint:
When taking the question, it would appear as:

What is encapsulation?

- The process of creating a variable that can be treated as any one of its subclasses.
- The process of structuring data types and functions into a higher level component.
- The process of creating subclasses that are hidden from the rest of program.
- The process of obtaining reusable methods and data fields from a parent object.
- The process of hiding the concrete implementation from the client. That way the client can focus on what is being processed rather than how it is being processed.

When grading the question, it would appear as:

What is encapsulation?

- The process of creating a variable that can be treated as any one of its subclasses.
- The process of structuring data types and functions into a higher level component.
- The process of creating subclasses that are hidden from the rest of program.
- The process of obtaining reusable methods and data fields from a parent object.
- The process of hiding the concrete implementation from the client. That way the client can focus on what is being processed rather than how it is being processed.

Comment: (given as feedback)

- No comment specified. -

Hint:

- No hint specified. -
Preview Question

When taking the question, it would appear as:

What is a "pure" object oriented language?

- Functions are treated as first-class citizens.
- Every type is treated consistently as an object.
- It supports only the object oriented paradigm.
- All computations are accomplished by message passing.
- The process of hiding the concrete implementation from the client. That way the client can focus on what is being processed rather than how it is being processed.

When grading the question, it would appear as:

What is a "pure" object oriented language?

   - Functions are treated as first-class citizens.
   - Every type is treated consistently as an object.
   - It supports only the object oriented paradigm.
   - All computations are accomplished by message passing.
   - The process of hiding the concrete implementation from the client. That way the client can focus on what is being processed rather than how it is being processed.

Comment: (given as feedback)

- No comment specified. -

Hint:

- No hint specified. -
When taking the question, it would appear as:

Dynamic binding is a key aspect to object oriented languages. It is what allows us to take advantage of polymorphism. How do we use dynamic binding in Python?

- Python uses dynamic binding by default. Nothing special needs to be done.
- The method must be marked as virtual in the base class, so the compiler will know to create a virtual method table. Python uses static binding by default.
- The base class must be an abstract class, so the compiler will know that all of its methods are virtual methods.
- The base class must be marked as dynamic, so the compiler knows to turn on the dynamic binding feature.
- The process of hiding the concrete implementation from the client. That way the client can focus on what is being processed rather than how it is being processed.

When grading the question, it would appear as:

Dynamic binding is a key aspect to object oriented languages. It is what allows us to take advantage of polymorphism. How do we use dynamic binding in Python?

- Python uses dynamic binding by default. Nothing special needs to be done.
- The method must be marked as virtual in the base class, so the compiler will know to create a virtual method table. Python uses static binding by default.
- The base class must be an abstract class, so the compiler will know that all of its methods are virtual methods.
- The base class must be marked as dynamic, so the compiler knows to turn on the dynamic binding feature.
- The process of hiding the concrete implementation from the client. That way the client can focus on what is being processed rather than how it is being processed.

Comment: (given as feedback)
- No comment specified.

Hint:
When taking the question, it would appear as:

Dynamic binding is a key aspect of object-oriented languages. It is what allows us to take advantage of polymorphism. How do we use dynamic binding in C++?

- C++ uses dynamic binding by default. Nothing special needs to be done.

- The method in the base class must be marked as virtual, so the compiler will know to create a virtual method table. C++ uses static binding by default.

- The base class must be marked as dynamic to turn on the dynamic binding feature.

- The base class must be an abstract data type, so the compiler knows that the methods are virtual methods.

- The process of hiding the concrete implementation from the client. That way the client can focus on what is being processed rather than how it is being processed.

When grading the question, it would appear as:

Dynamic binding is a key aspect of object-oriented languages. It is what allows us to take advantage of polymorphism. How do we use dynamic binding in C++?

- C++ uses dynamic binding by default. Nothing special needs to be done.

- The method in the base class must be marked as virtual, so the compiler will know to create a virtual method table. C++ uses static binding by default.

- The base class must be marked as dynamic to turn on the dynamic binding feature.

- The base class must be an abstract data type, so the compiler knows that the methods are virtual methods.

- The process of hiding the concrete implementation from the client. That way the client can focus on what is being processed rather than how it is being processed.

Comment: (given as feedback)

- No comment specified.

Hint:

https://ecat.montana.edu/d2l/common/popup/popup.d2l?ou=377992&queryString=ou%3D377992%26t%3D207903%26tid%3D2%26tid%3D51000%26act%3Dpr...
When taking the question, it would appear as:

If you were using Python, what would a proper "for loop" structure look like?

```python
for i in range(1, 100):
    print i
```

When grading the question, it would appear as:

If you were using Python, what would a proper "for loop" structure look like?

```python
for i in range(1, 100):
    print i
```
When taking the question, it would appear as:

Given the following pseudo-code for an object oriented language, what would be printed if the language used static binding?

```java
public class A{
    public int fun(int input){
        input = input*2;
        return input;
    }
}

public class B inherits from A{
    public int fun(int input){
        input = input*3;
        return input;
    }
}

public static void Main(String args[]){
    int x = 2;
    A a_var = new B();
    x = a_var.fun(x);
    print(x);
}
```

- 6
- The code would compile, but it would have runtime error.
- 4
- 12

When grading the question, it would appear as:

Given the following pseudo-code for an object oriented language, what would be printed if the language used static binding?

```java
public class A{
    public int fun(int input){
        input = input*2;
        return input;
    }
}

public class B inherits from A{
    public int fun(int input){
        input = input*3;
        return input;
    }
}

public static void Main(String args[]){
    int x = 2;
    A a_var = new B();
    x = a_var.fun(x);
    print(x);
}
```

When taking the question, it would appear as:

Given the following pseudo-code for an object oriented language, what would be printed if the language used dynamic binding?

```java
public class TheMaster{
    public wisdom(){
        print "Kung Fu is for self defense.";
    }
}

public class ThePupil inherits from TheMaster{
    public wisdom(){
        print "Kung Fu is for kicking butt.";
    }
}

public static void Main(String args[]){
    TheMaster zen = new ThePupil();
    zen.wisdom();
}
```

- The program would have a runtime error.
- Kung Fu is for self defense.
- Kung Fu is for kicking butt.
- The code would compile, but there is no output.

When grading the question, it would appear as:

Given the following pseudo-code for an object oriented language, what would be printed if the language used dynamic binding:

```java
public class TheMaster{
    public wisdom(){
        print "Kung Fu is for self defense.";
    }
}
```

- The program would have a runtime error.
- Kung Fu is for self defense.
- Kung Fu is for kicking butt.
- The code would compile, but there is no output.
When taking the question, it would appear as:

The output for the following Python code is:

class A:
    def m(self):
        print ("A")

class B(A):
    def m(self):
        print ("B")

class C(A):
    def m(self):
        print ("C")

class D(C,B):
    pass

a = D()
a.m()

- runtime error

- A

- B

**C**

When grading the question, it would appear as:

The output for the following Python code is:

class A:
    def m(self):
        print ("A")

class B(A):
    def m(self):
        print ("B")

class C(A):
    def m(self):
        print ("C")

```python
class D(C,B):
    pass
```
When taking the question, it would appear as:

Smalltalk supports multiple inheritance.

- True
- False

When grading the question, it would appear as:

Smalltalk supports multiple inheritance.

- True
- False

Comment: (given as feedback)

- No comment specified.

Hint:

- No hint specified.
When taking the question, it would appear as:

Java supports multiple inheritance.

- True
- False

When grading the question, it would appear as:

Java supports multiple inheritance.

- True
- False

Comment: (given as feedback)

- No comment specified.

Hint:

- No hint specified.
When taking the question, it would appear as:

C++ supports multiple inheritance.

- True
- False

When grading the question, it would appear as:

C++ supports multiple inheritance.

- True
- False

Comment: (given as feedback)

- No comment specified.

Hint:

- No hint specified.
Preview Question

When taking the question, it would appear as:

Python supports multiple inheritance.

☐ True
☐ False

When grading the question, it would appear as:

Python supports multiple inheritance.

⇒ True
False

Comment: (given as feedback)

- No comment specified. -

Hint:

- No hint specified. -
When taking the question, it would appear as:

What are the main principles of an Object Oriented language?

- First-class functions
- Formal logical structure
- Data abstraction
- Inheritance
- Encapsulation
- Procedural programming

When grading the question, it would appear as:

What are the main principles of an Object Oriented language?

- First-class functions
- Formal logical structure
- Data abstraction
- Inheritance
- Encapsulation
- Procedural programming

Comment: (given as feedback)

- No comment specified.

Hint:

- No hint specified.
When taking the question, it would appear as:

Now that we’ve covered many of the key design issues for object oriented programming languages, we could use our understanding to quickly familiarize ourselves with new languages.

By doing some research quickly online, select all the options that are true for C#.

- Static typing
- Dynamic typing
- Static binding
- Dynamic binding
- Strongly typed
- Weakly typed
- Multiple inheritance allowed
- Compiled
- Interpreted

When grading the question, it would appear as:

Now that we’ve covered many of the key design issues for object oriented programming languages, we could use our understanding to quickly familiarize ourselves with new languages.

By doing some research quickly online, select all the options that are true for C#.

- Static typing
- Dynamic typing
- Static binding
- Dynamic binding
- Strongly typed
When taking the question, it would appear as:

Continuing with C#, what would a proper "for loop" structure look like?

```csharp
for i:=0; i<100; i++
  fmt.Println(i)
}

for (int i=0; i<100; i++) {
  Console.WriteLine(i);
}

for i in range(1,100):
  print(i)

for i in 1..100 do
  puts "#{i}"
end
```

When grading the question, it would appear as:

Continuing with C#, what would a proper "for loop" structure look like?

```csharp
for i:=0; i<100; i++
  fmt.Println(i)
}

for (int i=0; i<100; i++) {
  Console.WriteLine(i);
}

for i in range(1,100):
  print(i)

for i in 1..100 do
  puts "#{i}"
end
```