Started on	Saturday, 15 August 2020, 10:47 AM
State	Finished
Completed on	Saturday, 15 August 2020, 10:54 AM
Time taken	7 mins 12 secs
Grade	**6.00** out of 10.00 (**60**%)

Question **1**

Correct

Mark 1.00 out of 1.00

Suppose the following three classes are defined:

```
public abstract class Figure {...}
public class Rectangle extends Figure {...}
public abstract class 3DFigure extends Figure {...}
```

Which of the following instantiations is *not* valid?

Select one:

- a. Rectangle r = new Rectangle(.....);
- b. Figure f = new Rectangle(....);
- c. Figure f = new 3DFigure(...); ✔

Your answer is correct.

The correct answer is: Figure f = new 3DFigure(...);

Question **2**

Correct

Mark 1.00 out of 1.00

The code below produces a compilation error. Examine the code and choose the fix that will enable the classes to compile:

```
public class Account {
    private double balance;
    public Account (double balance) { this.balance = balance; }
    // other getter and setter for balance
}
public class Savings extends Account {
    private double interestRate;
    public Savings(double rate) {
        this.interestRate = rate;
    }
}
```

Select one:

- a. Call the setBalance method of the Account from Savings
- b. Change the access of interestRate to public
- c. Add a no-arg constructor to class Savings
- d. Replace the constructor in Savings with one that calls the constructor of Account using super. ✔

Your answer is correct.

The correct answer is: Replace the constructor in Savings with one that calls the constructor of Account using super.

Question **3**
Incorrect
Mark 0.00 out of 1.00

Which of the following statements is *untrue* about method overriding?

Select one:

- a. Constructors cannot be overridden
- b. If a static method in the base class, is redefined in the sub-class, the later hides the method in the base class
- c. In method overriding, run-time polymorphism ensures that instantiated, the call to any method in the base class will be resolved to the correct method, based on the run-time type of the object instantiated. ✖
- d. During method overriding, the overridden method in the sub-class can specify a weaker access modifier

Your answer is incorrect.

The correct answer is: During method overriding, the overridden method in the sub-class can specify a weaker access modifier

Question **4**
Incorrect
Mark 0.00 out of 1.00

Choose the *incorrect* statement:

Select one:

- a. The principle of least knowledge reduces dependencies between objects and promotes loose coupling
- b. The code below is a good example of the principle of least knowledge

```
Driver driver = car.getDriver()
Address driverAddress = driver.getAddress()
```

- c. According to the principle of least knowledge, accessing the methods on objects returned by a method call is invalid
- d. The principle of least knowledge states that accessing methods of objects passed in as parameters or instantiated inside the method is valid ✖

Your answer is incorrect.

The correct answer is: The code below is a good example of the principle of least knowledge

```
Driver driver = car.getDriver()
Address driverAddress = driver.getAddress()
```

Question **5**
Incorrect
Mark 0.00 out of 1.00

Which of the following statements is FALSE in relation to generics?

Select one:

- a. A generic class used without type arguments is known as a *raw type*
- b. You cannot instantiate an array of a generic type using new operator e.g., T[] anArray = new T[100] ✖
- c. Consider the following method, someMethod(Box<Number> n) { /*.... */}, this method can take in a Box<Integer> or Box<Double>

Your answer is incorrect.

The correct answer is: Consider the following method, someMethod(Box<Number> n) { /*.... */}, this method can take in a Box<Integer> or Box<Double>

Question **6**
Correct
Mark 1.00 out of 1.00

A design pattern used to enhance a component's functionality dynamically at run-time is:

Select one:
- a. Composite Pattern
- b. Decorator Pattern ✔
- c. Abstract Factory Pattern
- d. Observer Pattern

Your answer is correct.

Question **7**
Correct
Mark 1.00 out of 1.00

Which Design Pattern should you use when....

- there is a language to interpret, and you can represent statements in the language as abstract syntax trees.

Select one:
- a. Singleton
- b. State
- c. Composite ✔
- d. Factory

Your answer is correct.

The correct answer is: Composite

Question **8**
Correct
Mark 1.00 out of 1.00

Which one of the following is not a code smell?

Select one:
- a. Classes that not only passively store data, but also methods to operate on the data ✔
- b. Large conditional logic blocks
- c. Methods making extensive use of another class

Your answer is correct.

The correct answer is: Classes that not only passively store data, but also methods to operate on the data

Question **9**
Correct
Mark 1.00 out of 1.00

Which of the following statements about design pattern is NOT true?

Select one:
- a. The collections.sort() method is a good example of the strategy Pattern
- b. The Java IO makes use of the composite pattern ✔
- c. The Java collection framework makes use of the Iterator Pattern

Your answer is correct.

The correct answer is: The Java IO makes use of the composite pattern

Question **10**

Incorrect

Mark 0.00 out of 1.00

An online camping store, sells different kinds of camping equipment. Items selected by the customer are added to a shopping cart. When a user clicks on the checkout Button, the application is required to calculate the total amount to be paid. The calculation logic for each item type varies, and we want to move all the calculation logic to one separate class, to decouple the different items from the calculation logic applied on them. As the application iterates through the disparate set of items of the shopping cart, we apply the price computation logic in the class to each item type. Which of the following patterns would be useful to design this scenario?

Select one:

- a. Strategy Pattern ✖
- b. Decorator Pattern
- c. Iterator Pattern
- d. Visitor Pattern

Your answer is incorrect.

The correct answer is: Visitor Pattern