CS390-2021-12A-12D(MY) >
↪☑ Tests & Quizzes

# Tests & Quizzes

Assessment Preview - This is an example student view of this assessment | done

# Quiz2-v3

Table of Contents

**Part 1 of 4** -

Question 1 of 25 1 Points

What will be the output for the given code?

```java
public interface Intface {
    default void myMethod(int x) {
        System.out.println(x);
    }
}


public interface SubIntface extends Intface {
}

public class SubImpl implements SubIntface{
    public static void main(String[] args) {
        SubImpl ob = new SubImpl();
        ob.myMethod(10);
    }
}
```

- ○ A. 10

- ○ B. Compile Error

- ○ C. Run time Error

Reset Selection

Question 2 of 25 1 Points

What is the output for the given code?

```java
public class SupClass {
    public void myMethod(int x) {
        System.out.println(x + 1);
    }
}

public interface SupInt1 {
    default void myMethod(int x) {
        myMethod(x + x);
    }
}

public class Implementer extends SupClass implements SupInt1 {
    public static void main(String[] args) {
        Implementer i = new Implementer();
        i.myMethod(3);
    }
}
```

- ○ A. 6
- ○ B. Compiler Error
- ○ C. Runtime Error
- ○ D. 4

Reset Selection

Question 3 of 25 1 Points

Choose whether the given code is valid or invalid in terms of Overriding?

```java
class Parent{
    void print(){
```

```
        System.out.println("Parent Class");
    }
}
```

```
public class Child extends Parent{

    @Override
    private void print(){
        System.out.println("Child Class");
    }

}
```

- ○ A. Valid

- ○ B. Compiler Error

- ○ C. Runtime Error

[Reset Selection](#)

Question 4 of 25 1 Points

What is the output of the given code?

```
class Fruit{
    Fruit(){
        System.out.println("Fruit");
    }
}
```

```
class Apple extends Fruit{
    Apple(){
        System.out.println("Apple");
    }
}
```

```
public class Sample {
    public static void main(String[] args) {
        Fruit ob = new Apple();
    }
}
```

- ○ A. Apple

- ○ B.

  Fruit

  Apple


- ○ C.

  Apple

  Fruit


- ○ D. Compiler Error

Reset Selection

Question 5 of 25 1 Points

What is causing an error in the following code?

```
protected class MyClass {
   private MyClass() {
      System.out.println("Inside MyClass constructor");
   }
   private class MyInnerClass {
      private MyInnerClass() {
         System.out.println("Inside MyInnerClass constructor");
      }
   }
}
```

- ○ A. Outer class cannot be protected

- ○ B. Inner class cannot be private

- ○ C. Outer class constructor cannot be private

- ○ D. Inner class constructor cannot be private

Reset Selection

Question 6 of 25 1 Points

A local inner class CAN be declared as which one of the following access modifiers.

- ○ A. private

- ○ B. protected

- ○ C. default (no modifier)

- ○ D. public

Reset Selection

Question 7 of 25 1 Points

Which one is CORRECTLY defined abstract class?

- ○ A. public class Device { public abstract void work(); }

- ○ B. public abstract class Device { public void work(); }

- ○ C. public abstract class Device { abstract void work(); }

- ○ D. public abstract class Device { public abstract void work() { System.out.println("Working"); } }

Reset Selection

Question 8 of 25 0 Points

Which one is NOT CORRECTLY defined interface?

- ○ A. interface Drivable{ void drive(); }

- ○ B. interface Drivable{ default void drive(){ System.out.println("driving..."); }; }

- ○ C. interface Drivable{ }

- ○ D. interface Drivable{ private abstract void drive(); }

Reset Selection

Question 9 of 25 1 Points

Can an object be a subclass of another object?

- ○ A. Yes---as long as single inheritance is followed.

- ○ B. No---inheritance is only between classes.

- ○ C. Only when one has been defined in terms of the other.

- ○ D. Yes---when one object is used in the constructor of another.

Reset Selection

Question 10 of 25 1 Points

What restriction is there on using the super reference in a constructor?

- ○ A. It can only be used in the parent's constructor.

- ○ B. Only one child class can use it.

- ○ C. It must be used in the last statement of the constructor.

- ○ D. It must be used in the first statement of the constructor.

Reset Selection

## Part 2 of 4 - True or False

Question 11 of 25 1 Points
Outer class can access members of the inner class directly.

○ A. True
○ B. False
Reset Selection

Question 12 of 25 1 Points
It is possible to create an object of abstract class.

○ A. True
○ B. False
Reset Selection

Question 13 of 25 1 Points
Anonymous inner classes can have any names.

○ A. True
○ B. False
Reset Selection

Question 14 of 25 1 Points

equals() and hashcode() should always be overridden along.

○ A. True

○ B. False

Reset Selection

Question 15 of 25 1 Points

Shallow copy is more secure than deep copy for object cloning.

○ A. True

○ B. False

Reset Selection

Question 16 of 25 1 Points. Point(s) deducted for incorrect answer: 1.0

Prior to java 8, all methods of an Interface are abstract, but abstract class does not have to have abstract methods.

○ A. True

○ B. False

Reset Selection

Question 17 of 25 1 Points. Point(s) deducted for incorrect answer: 1.0

Overloading method is polymorphism as its method is dynamic binding on runtime.

○ A. True

○ B. False

Reset Selection

Question 18 of 25 1 Points. Point(s) deducted for incorrect answer: 1.0

If you have nested loops with break in the inner loop, break can only terminate the inner enclosing loop.

○ A. True

○ B. False

Reset Selection

Question 19 of 25 1 Points. Point(s) deducted for incorrect answer: 1.0

In Java, a default constructor is the one which does not take any inputs, default constructors will be created by default in case you have no other constructor defined.

○ A. True

○ B. False

Reset Selection

Question 20 of 25 1 Points. Point(s) deducted for incorrect answer: 1.0
In Java, constructor chaining is the process of calling one constructor from another with respect to the current object. There could be any number of classes in the constructor chain.

○ A. True

○ B. False

Reset Selection

## Part 3 of 4 - Fill in the Blanks

Question 21 of 25 1 Points

An interface that contains just one abstract method is called a [                    ] interface

Question 22 of 25 1 Points

A [                    ] interface is an interface that has no methods or constants inside it. It provides run-time type information about objects, so the compiler and JVM have additional information about the object.

## Part 4 of 4 - Short answer questions.

Question 23 of 25 2 Points

Fill the blank spaces with correct codes to print the following result to the console.
----------------------------
Inner class value is: inner
Outer class value is: outer
----------------------------

```
public class Outer {
    public String value = "outer";
    public class Inner{
        public String value = "inner";
        void printValue() {
            System.out.println("Inner class value is: " + _____);
            System.out.println("Outer class value is: " + _____);
        }
    }

    public static void main(String[] args) {

        _____

        _____

    }
```

}

Maximum number of characters (including HTML tags added by text editor): 32,000
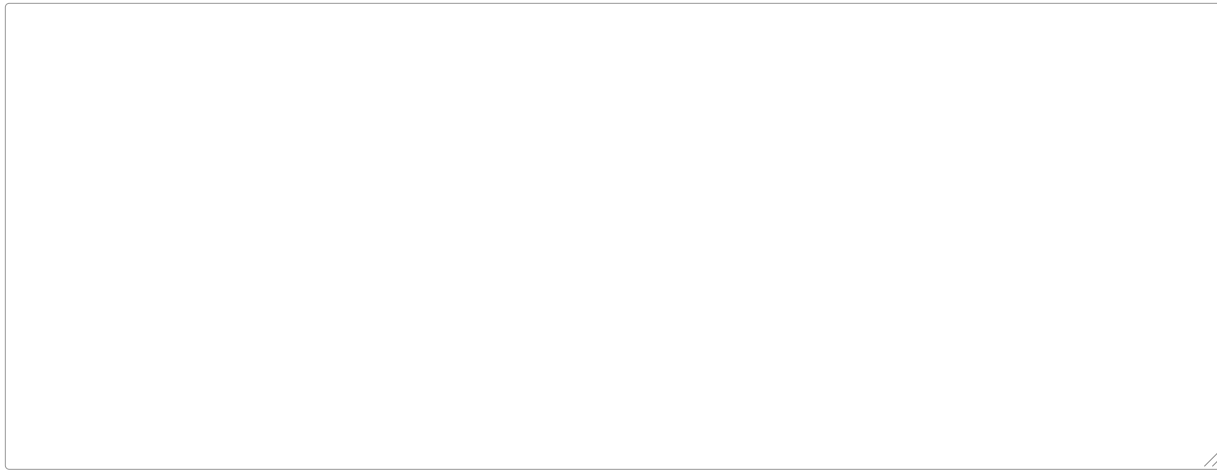
Question 24 of 25 1 Points

Implement showExample method with anonymous class and output "hello".

abstract class Example {
abstract void showExample();
}


public class Demo{
    public static void main(String[] args) {

        _____

    }
}

Maximum number of characters (including HTML tags added by text editor): 32,000

Question 25 of 25 5 Points

**Programming question**. Given an array of integers nums and an integer target, return indices of the two numbers such that they add up to target. You may assume that each input would have exactly one solution, and you may not use the same element twice. You can return the answer in any order.

Example 1:

Input: nums = [2,7,11,15], target = 9

Output: [0,1] Output: Because nums[0] + nums[1] == 9, we return [0, 1].

Example 2: Input: nums = [3,2,4], target = 6
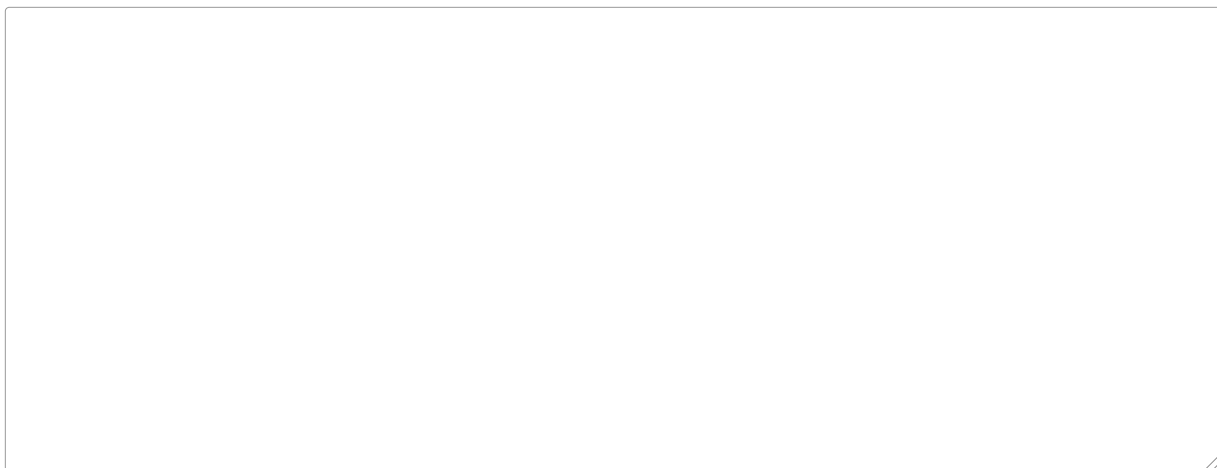
Output: [1,2]

Example 3: Input: nums = [3,3], target = 6

Output: [0,1]

Maximum number of characters (including HTML tags added by text editor): 32,000

Show Rich-Text Editor (and character count)

Save   Exit   Submit for Grading

Assessment Preview - This is an example student view of this assessment   done

- [Gateway](#)
- [Accessibility Information](#)
- [The Sakai Project](#)
- 🛈

  MUM Global Online Education 19.3

  ✖

          Tue, 30 Nov 2021 15:37:14 CST
  Server:
          Sakai3
  Build Info:
          RELEASE

Copyright 2003-2021 The Apereo Foundation. All rights reserved.

- Powered by Sakai

# Change Profile Picture

Error removing image
Error uploading image
Upload  Choose File   No file chosen

Save   Cancel   Connections   ✖                                                      Remove

Search for people ...

[View More](#)

My Connections    Pending Connections

You don't have any connnections yet. Search for people above to get started.

You have no pending connections.

←Back to My Connections

Search for people …

$({cmLoader.getString("connection_manager_no_results")}

Done