

✓ Congratulations! You passed!

TO PASS 80% or higher



GRADE 90%

Graded Quiz: Test your understanding of OOP in Java

LATEST SUBMISSION GRADE

90%

| 1. | Which of the following statements is false ? | 1/1 point |
|----|--|-----------|
| | The package java.lang is included with each Java application and does not need to be explicitly imported. | |
| | O You can import either individual classes in a package or all the objects in a package to use in your code. | |
| | Each Java package can only contain one object. | |
| | Packages create name spaces so that developers do not have to worry about other developers using the same name for their objects. | |
| | Correct Correct! Java packages can contain many objects, and they often do. | |
| | | |
| 2. | Which of the following statements about abstract classes is <u>true</u> ? | 1/1 point |
| | Abstract classes can only be inherited by ONE subclass. | |
| | Abstract classes cannot be instantiated and can only be inherited by subclasses. | |
| | Abstract classes cannot be instantiated or inherited by subclasses. They are used by directly referring to their class name. | |
| | Abstract classes can be instantiated but cannot be inherited by subclasses. | |
| | ✓ Correct | |

3. In the following code block, which method is an abstract method?

1/1 point

```
1 v public class Student {{
2
3
3
4
5
5
6
7 v public abstract void getGPA();
8
8
5 ystem.out.print("ID: " + studentID);
9
10
11 v public void setStudentID(String newID) {
12
13
14
15 v studentID = newID;
15
16
17
18
}
```

getStudent ID()

Correct!

- getGPA()
- setStudentID(String newID)
- printStudentID()

✓ Correct
Correct!

| public class Tools.circleleres (25.8); Tools area = Tools.circleleres (25.8); Tools area = Tools area | 4 4 5 6 6 6 7 7 8 8 8 O | <pre>public static float circleArea(float radius) { return radius * radius * pi; } 1 Tools tools = new Tools(); 2</pre> |
|--|-------------------------|---|
| Tools tools - new Tools.(r) | | return radius * radius * pi; } 1 Tools tools = new Tools(); 2 |
| Tools tools * new Tools(); 1 Tools area * tools.circle/res(25.8); 1 Tools area * Tools.static.circle/res(25.8); 1 Tools area * Tools.circle/res(25.8); 1 Tools area * Tools.circle/res(25.8); 1 Tools area * Tools.circle/res(25.8); 2 | | <pre>1 Tools tools = new Tools(); 2</pre> |
| ### Plant area - teols.circleArea(25.0); #### Plant area - Tools.static.circleArea(25.0); ################################### | | <pre>1 float area = Tools.circleArea(25.0); 1 float area = Tools.static.circleArea(25.0); 1 float area = (static)Tools.circleArea(25.0); 1 float area = Tools.circleArea(25.0);</pre> |
| ### Plant area - teols.circleArea(25.0); #### Plant area - Tools.static.circleArea(25.0); ################################### | | <pre>1 float area = Tools.circleArea(25.0); 1 float area = Tools.static.circleArea(25.0); 1 float area = (static)Tools.circleArea(25.0); 1 float area = Tools.circleArea(25.0);</pre> |
| I Float area = Tools.static.circleArea(25.0); | | <pre>1 float area = Tools.static.circleArea(25.0); 1 float area = (static)Tools.circleArea(25.0); 1 float area = Tools.circleArea(25.0);</pre> |
| ### Tools are a (static)Tools.circleArea(25.0); #### Tools are a Tools.circleArea(25.0); ################################### | | <pre>1 float area = (static)Tools.circleArea(25.0); 1 float area = Tools.circleArea(25.0);</pre> |
| ### Tools are a (static)Tools.circleArea(25.0); #### Tools are a Tools.circleArea(25.0); ################################### | | <pre>1 float area = (static)Tools.circleArea(25.0); 1 float area = Tools.circleArea(25.0);</pre> |
| The following code block defined a static variable pl. What is the correct way to use this variable in our code? 1 | | 1 float area = Tools.circleArea(25.0); Correct |
| The following code block defined a static variable pl. What is the correct way to use this variable in our code? 1 | | 1 float area = Tools.circleArea(25.0); Correct |
| The following code block defined a static variable pl. What is the correct way to use this variable in our code? * public class Tools { | | ✓ Correct |
| The following code block defined a static variable pl. What is the correct way to use this variable in our code? * public class Tools { | | ✓ Correct |
| The following code block defined a static variable pl. What is the correct way to use this variable in our code? * public class Tools { | | ✓ Correct |
| The following code block defined a static variable pi. What is the correct way to use this variable in our code? **public class Tools { | ~ | |
| The following code block defined a static variable pi. What is the correct way to use this variable in our code? **public class Tools { | ~ | |
| The following code block defined a static variable pi. What is the correct way to use this variable in our code? **public class Tools { | ~ | |
| The following code block defined a static variable pi. What is the correct way to use this variable in our code? **public class Tools { | ~ | |
| The following code block defined a static variable pi. What is the correct way to use this variable in our code? *** public class Tools !{ | | Correct! |
| ** public class Tools {{ | | |
| ** public class Tools {{ | | |
| ** public class Tools {{ | | |
| public static float pi = 3.1417; public static float circleArea(float radius) { return radius * radius * pi; l Tools.static.pi 1 (static float)Tools.pi 1 pi 1 Tools.pi Verrect Correct Correct Which of the follow statements regarding final methods is true? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect Refer to Task 7 if you need a refresher. | . The f | following code block defined a static variable pi . What is the correct way to use this variable in our code? |
| public static float circleArea(float radius) { return redius * radius * pi; } 1 Tools.static.pi 1 Tools.pi 1 Tools.pi Which of the follow statements regarding final methods is true? When a method is marked final, no subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. I Incorrect Incorrect Refer to Task 7 if you need a refresher. | | |
| public static float circleArea(float radius) { return radius * radius * pi; } 1 Tools.static.pi 1 [static float)Tools.pi 1 pi 1 Tools.pi Which of the follow statements regarding final methods is true? Which of the follow statements regarding final methods is true? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect Refer to Task 7 if you need a refresher. | 3 | public static float pi = 3.1417; |
| 1 Tools.static.pi 1 (static float)Tools.pi 1 pi 1 Tools.pi 2 Tools.pi 3 Tools.pi 4 Tools.pi 5 Tools.pi 6 Which of the follow statements regarding final methods is true? 6 When a method is marked final, no subclasses can access it. 6 When a method is marked final, only the object itself or its subclasses can access it. 6 When a method is marked final, no subclasses may override and redefine it. 7 When a method is marked final, it will not have a function body. 8 1 Tools.static.pi 2 Tools.static.pi 3 Tools.static.pi 1 Tools.static.pi 2 Tools.pi 3 Tools.pi 4 Tools.pi 5 Tools.pi 6 Tools.static.pi 6 Tools.pi 7 Tools.pi 8 Tools.pi 9 Tools.pi 1 Tools.pi 2 Tools.pi 3 Tools.pi 4 Tools.pi 5 Tools.pi 6 Tools.pi 1 Tools.pi 1 Tools.pi 1 Tools.pi 1 Tools.pi 2 Tools.pi 1 Tools.pi 2 Tools.pi 3 Tools.pi 1 Tools.pi 1 Tools.pi 1 Tools.pi 1 Tools.pi 1 Tools.pi 2 Tools.pi 2 Tools.pi 3 Tools.pi 4 Tools.pi 5 Tools.pi 6 Tools.pi 1 | 5 | <pre>public static float circleArea(float radius) {</pre> |
| 1 Tools.static.pi 1 | 7 | } |
| 1 | 8 | |
| 1 | | 1 Tools statis al |
| □ 1 Tools.pi ☑ 1 Tools.pi ✓ Correct Correct! Which of the follow statements regarding final methods is true? ⑥ When a method is marked final, no subclasses can access it. ○ When a method is marked final, only the object itself or its subclasses can access it. ○ When a method is marked final, no subclasses may override and redefine it. ○ When a method is marked final, it will not have a function body. ! Incorrect Incorrect Refer to Task 7 if you need a refresher. | 0 | 1 1001S.Static.pi |
| □ 1 Tools.pi ☑ 1 Tools.pi ✓ Correct Correct! Which of the follow statements regarding final methods is true? ⑥ When a method is marked final, no subclasses can access it. ○ When a method is marked final, only the object itself or its subclasses can access it. ○ When a method is marked final, no subclasses may override and redefine it. ○ When a method is marked final, it will not have a function body. ! Incorrect Incorrect Refer to Task 7 if you need a refresher. | | |
| □ 1 Tools.pi ☑ 1 Tools.pi ✓ Correct Correct! Which of the follow statements regarding final methods is true? ⑥ When a method is marked final, no subclasses can access it. ○ When a method is marked final, only the object itself or its subclasses can access it. ○ When a method is marked final, no subclasses may override and redefine it. ○ When a method is marked final, it will not have a function body. ! Incorrect Incorrect Refer to Task 7 if you need a refresher. | | 1 (static float)Tools.ni |
| Tools.pi Tools.pi Verrect Correct! Which of the follow statements regarding final methods is true? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect Refer to Task 7 if you need a refresher. | | 1 (30022 12007)0023.92 |
| Tools.pi Tools.pi Verrect Correct! Which of the follow statements regarding final methods is true? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect Refer to Task 7 if you need a refresher. | | |
| ✓ Correct Correct! Which of the follow statements regarding final methods is true? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect Incorrect! Refer to Task 7 if you need a refresher. | 0 | 1 pi |
| ✓ Correct Correct! Which of the follow statements regarding final methods is true? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | |
| ✓ Correct Correct! Which of the follow statements regarding final methods is true? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | |
| Which of the follow statements regarding final methods is true ? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | • | 1 Tools.pi |
| Which of the follow statements regarding final methods is true ? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | |
| Which of the follow statements regarding final methods is true ? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | |
| Which of the follow statements regarding final methods is true ? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | |
| Which of the follow statements regarding final methods is true ? When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | ~ | / Correct |
| When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | Correct! |
| When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | |
| When a method is marked final, no subclasses can access it. When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | |
| When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | . Whic | h of the follow statements regarding final methods is <u>true</u> ? |
| When a method is marked final, only the object itself or its subclasses can access it. When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | When a method is marked final, no subclasses can access it. |
| When a method is marked final, no subclasses may override and redefine it. When a method is marked final, it will not have a function body. Incorrect Incorrect! Refer to Task 7 if you need a refresher. | | |
| When a method is marked final, it will not have a function body. ! Incorrect Incorrect! Refer to Task 7 if you need a refresher. | 0 | When a method is marked final, only the object itself or its subclasses can access it. |
| Incorrect! Refer to Task 7 if you need a refresher. | 0 | When a method is marked final, no subclasses may override and redefine it. |
| Incorrect! Refer to Task 7 if you need a refresher. | | When a method is marked final, it will not have a function body. |
| Incorrect! Refer to Task 7 if you need a refresher. | | mien a meanda is markea imar, it wiir not have a function body. |
| Incorrect! Refer to Task 7 if you need a refresher. | | |
| | ! | |
| Which of the following statements about abstract methods is true ? | | Incorrect! Refer to Task 7 if you need a refresher. |
| Which of the following statements about abstract methods is true? | | |
| Which of the following statements about abstract methods is true? | | |
| | . Whic | h of the following statements about abstract methods is <u>true</u> ? |
| An abstract method declaration can have a function be during to the substance and the | | An abstract method declaration can have a function hads but sub-large and the |
| An abstract method declaration can have a function body but subclasses can override it. | 0 | an auxuract metrioù declaration can nave a l'unction dody dut sudclasses can override it. |
| An abstract method must also be declared final . | 0 | An abstract method must also be declared final . |
| An abstract method declaration does not have a function body because the subclasses will define the function | | An abstract method declaration does not have a function body because the subclasses will define the function |

| | An abstract method must not return any value. | |
|-----|---|-----------|
| | ✓ Correct Correct! | |
| 8. | One of the purposes of Java interface is: | 1/1 point |
| | Provides a way to present data to the user. | |
| | Allow objects to communicate with each other directly. | |
| | Allow different programming languages to work together. | |
| | Specify what methods need to be defined for the objects that implement that interface. | |
| | ✓ Correct Correct! | |
| | | |
| 9. | Regarding this code block, which of the statements is <u>true</u> ? | 1/1 point |
| | <pre>1 public static final boolean checkPassword(String password) { 2 boolean result = false; 3 // logic to check password here 4 return result; 5 }</pre> | |
| | The method checkPassword is a static method and it cannot be overridden in any subclasses. | |
| | The method checkPassword() is a static method but becomes non-static if it is overridden and redefined in its subclasses. | |
| | Will generate an error when compiled because static and final cannot be used together. | |
| | The method checkPassword() cannot be used unless it is overridden and redefined in its subclasses. | |
| | ✓ Correct Correct! | |
| 10. | . Which of the following statements about Java interface is <u>true</u> ? | 1/1 point |
| | Methods in interfaces have function bodies although they can be overridden by objects that implement them. | |
| | Abstract classes cannot declare implementation of interfaces. | |
| | An object can choose to implement more than one interface. | |
| | Interfaces are properties of objects and only their subclasses can implement those interfaces. | |
| | ✓ Correct Correct! We did in task 9! | |