

Review Test Submission: Test02

User	Yolanda Gunter
Course	Object-Oriented Programming (Java) (91199)
Test	Test02
Started	10/6/24 11:04 PM
Submitted	10/6/24 11:55 PM
Due Date	10/6/24 11:59 PM
Status	Completed
Attempt Score	92 out of 100 points
Time Elapsed	51 minutes
Results Displayed	All Answers, Submitted Answers, Correct Answers, Feedback, Incorrectly Answered Questions

Question 1

4 out of 4 points



Instance variables and *instance* methods can only be accessed through an object of the class in Java.

Selected Answer: ☒ TrueAnswers: ☒ True
☐ False

Question 2

4 out of 4 points



A value of the type **int** may not have a fractional part.

Selected Answer: ☒ TrueAnswers: ☒ True
☐ False

Question 3

4 out of 4 points



The syntax for a class definition in Java requires a semicolon following the closing curly bracket.

Selected Answer: ☒ False

Answers: True
☒ False

Question 4

4 out of 4 points



When using command-line arguments in Java, if the name of the string array is **args**, the **args.length** attribute can be used by the code in the program to determine the number of arguments actually entered.

Selected Answer: ☒ True
Answers: ☒ True
False

Question 5

4 out of 4 points



At any instant in time, the *state* of an object is determined by the values stored in its methods and its behavior is determined by its instance variables.

Selected Answer: ☒ False
Answers: True
☒ False

Question 6

4 out of 4 points



The **println** method can be accessed by joining the name of a variable that references a **PrintStream** object to the name of the **println** method using a period as the joining operator.

Selected Answer: ☒ True
Answers: ☒ True
False

Question 7

0 out of 4 points



The first string in the array of command-line arguments in the main method contains the name of the Java application

Selected Answer: ☒ True
Answers: True
☒ False

Response Feedback: Unlike C++, the first string in the array of command-line arguments in the main method does **not** contain the name of the application.

Question 8

4 out of 4 points



In Java, the class name alone is sufficient for accessing *class* variables and *class* methods by joining the name of the class with the name of the variable or method using a colon.

Selected Answer: ☒ False

Answers: ☐ True
☒ False

Question 9

4 out of 4 points



If the Java application is not designed to use command-line arguments, it is not necessary to include a formal argument list for the **main** method.

Selected Answer: ☒ False

Answers: ☐ True
☒ False

Question 10

4 out of 4 points



The **main** method cannot access the variables and methods of objects instantiated from other classes.

Selected Answer: ☒ False

Answers: ☐ True
☒ False

Question 11

4 out of 4 points



The **println** method is an instance method of the System class.

Selected Answer: ☒ False

Answers: ☐ True
☒ False

Question 12

4 out of 4 points



The **main** method in the controlling class of a Java application controls the flow of the program.

Selected Answer: ☒ True

Answers: ☒ True
☐ False

Question 13

4 out of 4 points



A value of the type **int** may have a fractional part.

Selected Answer: ☒ False

Answers: ☐ True
☒ False

Question 14

4 out of 4 points



In order for the **System** class to be useful to an application, the application must instantiate objects of the **System** class.

Selected Answer: ☒ False

Answers: ☐ True
☒ False

Question 15

4 out of 4 points



The **println** method causes its argument to be displayed on the standard output device.

Selected Answer: ☒ True

Answers: ☒ True
☐ False

Question 16

4 out of 4 points



The **main** method must instantiate objects of other classes in order for the program to execute.

Selected Answer: ☒ False

Answers: ☐ True
☒ False

Question 17

4 out of 4 points



The **static** keyword when used as the return type for any Java method indicates that the method does not **return** anything.

Selected Answer: ☒ False

Answers: ☐ True

☒ False

Question 18

4 out of 4 points



Class variables such as the **out** variable of the **System** class must be of some specific type.

Selected Answer: ☒ True

Answers: ☒ True

☐ False

Question 19

0 out of 4 points



Class variables must be of a primitive type such as **int** or **float**.

Selected Answer: ☒ True

Answers: ☐ True

☒ False

Response Feedback: A *class* variable can be a primitive type, or it can be a reference variable that points to another object.

Question 20

4 out of 4 points



An object of type **PrintStream** is automatically instantiated when the **System** class is loaded into an application.

Selected Answer: ☒ True

Answers: ☒ True

☐ False

Question 21

4 out of 4 points



The **out** variable in the **System** class refers to an instance of the **PrintStream** class (a *PrintStream* object), which is automatically instantiated when the **System** class is loaded into the application.

Selected Answer: ☒ True

Answers: ☒ True

False

Question 22

4 out of 4 points



The reason that the **main** method in a Java application must be declared **static** is because the **static** keyword indicates that the method is a *class* method which can be called without the requirement to instantiate an object of the class. This is used by the Java virtual machine to launch the program by calling the **main** method of the class identified in the command to start the program.

Selected Answer: ☒ TrueAnswers: ☒ True
☐ False**Question 23**

4 out of 4 points



The **out** variable in the **System** class is of a primitive type.

Selected Answer: ☒ FalseAnswers: ☐ True
☒ False**Question 24**

4 out of 4 points



Given that the *controlling class* of every Java application must contain a **main** method, other classes in the program may not also have a **main** method

Selected Answer: ☒ FalseAnswers: ☐ True
☒ False**Question 25**

4 out of 4 points



The void keyword is used to cause a variable or method to become a *class* variable or *class* method in Java

Selected Answer: ☒ FalseAnswers: ☐ True
☒ False

Sunday, October 6, 2024 11:55:15 PM CDT

← OK