Quiz 2: Methods

# Question 1: Method Overload

Consider the following Java code:

public class punchcard {  
 public static int add(int a, int b) {  
 return a + b;  
 }  
 public static double add(double a, double b) {  
 return a + b;  
 }  
 public static void main(String[] args) {  
 int result1 = add(2, 3);  
 double result2 = add(2.5, 3.7);  
 System.out.println(result1);  
 System.out.println(result2);  
 }  
}

What would be the output of the code?

1. 5 and 6
2. 5 and 6.2
3. 5.0 and 6
4. 5.0 and 6.2
5. error

*answer*: b

# Question 2: Recursive Method

Consider the following recursive Java method:

The factorial method should break out of the recursion when the value of n is 0 or 1. Otherwise, it should return the product of n and the result of calling factorial with n - 1.

public class punchcard {  
 public static int factorial(int n) {  
 if (n == 0 || n == 1) {  
 \_\_\_\_\_\_\_\_;  
 } else {  
 return n \* factorial(n - 1);  
 }  
 }  
  
 public static void main(String[] args) {  
 int result = factorial(4);  
 System.out.println(result);  
 }  
}

What should be placed instead of blank?

1. break
2. main()
3. return
4. return 0
5. return 1

*answer*: d

# Question 1: Method Pass-by-Value

Consider the following Java code:

public class PunchCard {  
 public static void modifyValue(int x) {  
 x = x \* 2;  
 }  
 public static void main(String[] args) {  
 int num = 5;  
 modifyValue(num);  
 System.out.println(num);  
 }  
}

What will be the value of num after the modifyValue method is called?

1. Null
2. 0
3. 5
4. 10
5. The code will result in a compilation error.

*answer*: c

# Question 6: Method Overload with Strings

Consider the following Java code:

public class StringOverloadExample {  
 public static void displayMessage(String message) {  
 System.out.println("Message: " + message);  
 }  
 public static void displayMessage(String message, int times) {  
 for (int i = 0; i < times; i++) {  
 System.out.println("Message: " + message);  
 }  
 }  
 public static void main(String[] args) {  
 displayMessage("Hello");  
 displayMessage("Java", 3);  
 }  
}

Fill the blank so the second method that print the message multiple times will be called.

1. displayMessage("Java", 3);
2. displayMessage("Java");
3. displayMessage.second("Java");
4. displayMessage(String "Java",int 3);
5. It’s not possible to call the second method. The java does not allow multiple methods with the same name.

*answer*: a