Self Quiz Unit 3 - a mixture of three different self quizzes

Programming 1 (University of the People)

1. Question text

Consider the following Java method, which term best describes "'("Hello, World!")"?

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
```

Select one:

○ a. actual parameter or argument

○ b. formal parameter

○ c. method call

○ d. modifier

● e. return type

Feedback

Your answer is incorrect.

See Section 4.3.2 of Eck (2014).

The correct answer is: actual parameter or argument

2. Question text

Consider the following Java method, which term best describes "String[] args"?

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
```

Select one:

○ a. actual parameter or argument

● b. formal parameter

○ c. method call

○ d. modifier

○ e. return type

Feedback

Your answer is correct.

See Section 4.3.2 of Eck (2014).

The correct answer is: formal parameter

3. Question text

Which one of the following terms does NOT describe a desirable interface to a black box?

Select one:

● a. easy to understand

○ b. implementation

○ c. public

○ d. specification

○ e. straightforward

Feedback

Your answer is incorrect.

The correct answer is: implementation

4. Question text

Which one of the following is NOT part of the signature of a Java method?

Select one:

○ a. method name

○ b. names of formal parameters

● c. number of formal parameters

○ d. types of formal parameters

Feedback

Your answer is incorrect.

The names of formal parameters are only important for the implementation of the method. See Section 4.3.3 of Eck (2014).

The correct answer is: names of formal parameters

5. Question text

What is the output of the following Java program?

```
class Sum {
    static int sum = 0;
    static void add() { sum++; }
    public static void main(String[] args) {
        for (int i = 0; i < 10; i++) add();
        System.out.println(sum);
    }
}
```

Select one:

○ a. 0

● b. 9

○ c. 10

○ d. 45

○ e. 100

Your answer is incorrect.

The "add" method directly increments the static member "sum" ten times. See Section 4.2.4 of Eck (2014).

The correct answer is: 10

6. Question text

What is the output of the following Java program?

```
class Sum {
    static int sum = 0;
    static void add(int i) { sum += i; }
    public static void main(String[] args) {
        for (int i = 0; i < 10; i++) add(i);
        System.out.println(sum);
    }
}
```

Select one:

● a. 0

○ b. 9

○ c. 10

○ d. 45

○ e. 100

Feedback

Your answer is incorrect.

The "add" method adds the value of its parameter to the static member variable "sum". Thus "sum" gets the value 0+1+2+3+4+5+6+7+8+9, or 45. See Sections 4.2.4 and 4.3.2 of Eck (2014).

The correct answer is: 45

7. Question text

Consider the following first line from a Java method definition.

`public static boolean compute(int n, float x) {`

Which one of the following lines could begin a method that legally overloads the above method?

Select one:

○ a. private static boolean compute(int n, float x) {

○ b. public boolean compute(int n, float x) {

○ c. public static int compute(int n, float x) {

● d. public static boolean compute(float n, int x) {

○ e. public static boolean compute(int x, float n) {

Feedback

Your answer is incorrect.

To overload a method, a method must have the same name but different parameter types (or a different number of parameters). See Section 4.3.3 of Eck (2014).

The correct answer is: public static boolean compute(float n, int x) {

8. Question text

Consider the following Java method, which term best describes "void"?

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
```

Select one:

○ a. actual parameter or argument

○ b. formal parameter

○ c. method call

○ d. modifier

● e. return type

Feedback

Your answer is correct.

See Section 4.4 of Eck (2014).

The correct answer is: return type

9. Question text

What is output by the following Java program?

```
Zap {
    static boolean zap() { return true; }
    static int zap(boolean x) { return 0; }
    static double zap(int x) { return 0.5; }
    static String zap(double x) { return "Zap!"; }
    static boolean zap(String x) { return false; }
    public static void main(String[] args) {
        System.out.println(zap(zap(zap(zap(1)))));
    }
}
```

Select one:

○ a. true

○ b. 0

○ c. 0.5

● d. Zap!

○ e. false

Feedback

Your answer is incorrect.

The innermost call to "zap" has argument 1, which is an int, so it uses the third method definition, which returns the double 0.5. The next "zap" call then gets this double as an argument, so it uses the fourth method definition, which returns the String "Zap!" So the next call gets a String argument, which means it uses the fifth method definition, which returns the boolean false. The outermost "zap" call gets this boolean, so it uses the second definition and returns the int 0. The first definition of "zap" is never used. See Section 4.3.3 of Eck (2014).

The correct answer is: 0

Question 5

Incorrect

Mark 0.00 out of 1.00

Flag question

Question text

What is output by the following Java program?

```
class Zap {
    static boolean zap() { return true; }
    static int zap(boolean x) { return 0; }
    static double zap(int x) { return 0.5; }
    static String zap(double x) { return "Zap!"; }
    static boolean zap(String x) { return false; }
    public static void main(String[] args) {
        System.out.println(zap(zap(zap(zap()))));
    }
}
```

Select one:

● a. true

○ b. 0

○ c. 0.5

○ d. Zap!

○ e. false

Feedback

Your answer is incorrect.

The innermost call to "zap" has no argument, so it uses the first method definition, which returns the boolean value true. The next call to "zap" thus gets a boolean argument, so it uses the second method definition, which returns the int 0. The next call to "zap" gets this int argument, so it uses the third method definition, which returns the double 0.0. The outmost call to "zap" thus gets a double argument, so it uses the fourth method definition, which returns the String "Zap!" See Section 4.3.3 of Eck (2014).

The correct answer is: Zap!

Question 1

Incorrect

Mark 0.00 out of 1.00

Flag question

Question text

What is output by the following Java program?

```
class Compute {
    static int compute() { return 42; }
    static int compute(int i) { return i+1; }
    public static void main(String[] args) {
        System.out.println(compute(compute()));
    }
}
```

Select one:

○ a. 0

○ b. 1

○ c. 2

● d. 42

○ e. 43

Feedback

Your answer is incorrect.

The inner call to "compute()" has no argument, so it uses the first definition of the method, which returns the value 42. This 42 becomes the argument to the outer call. Because that call does have an argument, it uses the second definition of "compute". This method then returns 42+1, or 43. See Section 4.3.3 of Eck (2014).

The correct answer is: 43

Question 2

Incorrect

Mark 0.00 out of 1.00

Flag question

Question text

Consider the following Java method, which term best describes "public"?

```
public static void main(String[] args) {
    System.out.println("Hello, World!");
}
```

Select one:

○ a. actual parameter or argument

○ b. formal parameter

● c. method call

○ d. modifier

○ e. return type

Feedback

Your answer is incorrect.

See Section 4.2.1 of Eck (2014).

The correct answer is: modifier

Question 3

Incorrect

Mark 0.00 out of 1.00

Flag question

Question text

Where in a Java file can the text "int x" NOT appear?

Select one:

● a. in a class definition but outside a method definition

○ b. in a for statement

○ c. in a method call

○ d. in a method definition

○ e. in a method parameter list

Feedback

Your answer is incorrect.

Method calls have variable names, values, and expressions. The do not have type identifiers. See Sections 4.2.2, 4.2.4, 4.3.2, and 4.7.1 of Eck (2014).

The correct answer is: in a method call

Question 4

Incorrect

Mark 0.00 out of 1.00

Flag question

Question text

Which one of the following lines is a Java method call?

Select one:

○ a. import javax.swing.JOptionPane;

○ b. int x = 0;

○ c. compute();

○ d. return y;

● e. void compute() {}

Feedback

Your answer is incorrect.

See Section 4.2.2 of Eck (2014).

The correct answer is: compute();

Question 5

Incorrect
Mark 0.00 out of 1.00

Flag question

Question text

A Java method gets executed when it is...

Select one:

○ a. called

● b. compiled

○ c. declared

○ d. defined.
defined

○ e. imported

Feedback

Your answer is incorrect.

See Section 4.2.2 of Eck (2014).

The correct answer is: called