OOP Java Exercise 3 Task 3 - Quizes

1) Given:

```
-------------------------------------------------
public class Hello< T1 , T2>          {
        public T1       output1;
        public T2       output2;
        public T1       getOutput1( ){
                return output1;         }
        public T2       getOutput1( ){
                return output2;         }
        public void     talk( ) {
                System.out.println( getOutput1 + " " + getOutput2);
                return;          }
}
-------------------------------------------------
```

Which of the following can be used to print "Hello World" ?

a) Hello<String, Double> h = new Hello<String, Double>("Hello World" , 4.2)
h.talk();

b) Hello<String, Double> h = new Hello<>("Hello World" , 4.2)
h.talk();

c) Hello<String, String> h = new Hello<String, String>("Hello" , "World")
h.talk();

d) Hello<String, Double> h = new Hello<>("Hello" , "World")
h.talk();

e) Hello<String, String> h = new Hello<String, String>("Hello World" , "")
h.talk();

2) What can wildcards be used as?

a) type argument

b) type of a parameter

c) local variable

d) supertype

3) How many different type arguments can a generic class have?

a) 1

b) 8

c) 32

d) any

4) Given:

```
-------------------------------------------------
public class A <T>   {
        ...
}
static public void main(string[] args)        {
        A <Integer> a = new <Integer> A( );
        ...
}
-------------------------------------------------
```

T in "public class A <T>" is called a ... ?

a) type
b) type variable
c) type argument
d) type parameter
e) type declaration

5) Given:

```
-------------------------------------------------
public void tripleNumber(Number n)       {
        System.out.println(n*3);
}
-------------------------------------------------
```

Which of the following can be used to print 9

a)
```
int i = 3;
tripleNumber(i);
```
b) tripleNumber(new Double(3);
c) tripleNumber(new String („3“));
d) tripleNumber(new boolean(true);

6) How are generic objects created in Java?

a) Test object = new Test();
b) Test<Integer> = new Test<Integer>();
c) Test<String> = new Test<>();

7) How are generic methods in Java declared?

a) public static <T> boolean test(T p1, T p2)
b) public boolean test(<T> p1)
c) private interface test(T p1)

8) Given:

```
-------------------------------------------------
public class Value< T>       {
        public T        value;
        public Value<T>(T t)        {
                value = t;              }
        public T        getValue( )     {
                return value;           }
        public void     setValue(T t) {
                value = t;
                return;                 }
        public void     showValue( ) {
                System.out.println( getValue);
                return;                 }
}
-------------------------------------------------
```

Which of the following can be used to output „10“

a) Value<int> v = new Value<>(10);
v.showValue();

b) Value<int> v1 = new Value<int>(10);
Value v2 = v1;
v2.showValue();

c) Value<int> v1 = new Value<int>(10);
Value<string> v2 = v1;
v2.showValue();

9) Which paramater naming is extensively used by the Java Collections Framework?

a) T - Type

b) V - Value

c) E - Element

d) K - Key

e) N - Number

10) Why use generics in Java?

a) Stronger type checks at compile time.

b) Elimenation of casts.

c) To write generic code.

d) To use less RAM.

11) What are bounded type parameters used for?

a) Restricting the the value of numbers

b) Restricting the types that can be used as arguments

c) To bind certain type parameters to a specific type

12) What does generics in Java enable?

a) Variables

b) Types (classes and interfaces)

c) Lists

d) Arrays

13) How is a generic class defined in Java?

a) class name

b) name class

c) class name<T1, T2, ..., Tn>

14) Why is there a parameter naming convention for generic classes in Java?

a) It would be difficult to tell generic and non generic classes and interfaces apart.

b) It makes the code run faster.

c) It makes the code easier to read.

15) How can you subtype a generic class or interface in Java?

a) extendig

b) implementing

c) including

d) excluding