1. When you create an instance of a generic class, what types can you pass as arguments to the class's type parameter?

a. primitive types only

b. reference types only

c. interface types only

d. primitive, reference, and interface types

2. In a generic method, a type parameter is defined_____.

a. inside the parentheses, along with the method's parameter variables

b. after the method's return type, and before the method's name

c. before the method's return type

d. inside the body of the method, but before any local variables are declared

3. Look at the following method header:

void displayPoint(Point <Number> myPoint)

Which of the following objects would we be allowed to pass as an argument to the displayPoint method? (Select all that apply.)

a. Point <Number> p;

b. Point <Integer> p;

c. Point <Double> p;

d. Point <String> p;

4. Look at the following method header:

void displayPoint(Point<?> myPoiny myPoint)

Which of the following objects would we be allowed to pass as an argument to the displayPoint method? (Select all that apply.)

a. Point <Number> p;

b. Point <Integer> p;

c. Point <Double> p;

d. Point <String> p;

5. Look at the following method header:

void displayPoint(Point <? Extends Number> myPoint)

Which of the following objects would we be allowed to pass as an argument to the displayPoint method? (Select all that apply.)

a. Point <Number> p;

b. Point <Integer> p;

c. Point <Double> p;

d. Point <String> p;

6. Look at the following method header:

void displayPoint(Point <? Super Double> myPoint)

Which of the following objects would we be allowed to pass as an argument to the displayPoint method? (Select all that apply.)

a. Point <Number> p;

b. Point <Integer> p;

c. Point <Double> p;

d. Point <String> p;

7. In the generic-type notation <T extends Number> to what type of bound is T constrained?

a. an upper bound

b. a lower bound

c. both an upper bound and lower bound

d. T is not constrained to a bound

8. In the generic-type notation <T super Integer> to what type of bound is T constrained?

a. an upper bound

b. a lower bound

c. both an upper bound and lower bound

d. T is not constrained to a bound

9. Which of the following generic type notations uses a wildcard?

a. Point<W>

b. Point<T extends Number>

c. Point<T super Integer>

d. Point<?>

10. The process used by the Java compiler to remove generic notation and substitute an actual type for type parameters is known as_____.

a. erasure

b. removal

c. substitution

d. masking

11. **True or False**: It is better to discover an error at runtime than at compile time.

12. **True or False**: It is possible to instantiate a generic class without specifying a type argument.

13. **True or False**: Type parameters must be single character identifiers, written in uppercase.

14. **True or False**: You can constrain a type parameter to both an upper bound and lower bound.

15. **True or False**: A generic class can extend a non-generic class.

16. **True or False**: You cannot create an array of generic class objects.

17. **True or False**: A generic class's type parameter can be the type of a static field.

18. **True or False**: An exception class cannot be generic.

Algorithm Workbench

1.

```
ArrayList<Customer> customers = new ArrayList<>();
```

2.

```
12
13      ArrayList<String> names = new ArrayList<>();
14
15      names.add(e: "William Jefferson");
16
```

3.

```
2
3   public class MyType <T extends String > {
4
5
```

4.

```
2
3   public class MyType <T super Integer > {
4
5
```

5.

```java
public static <T extends Comparable<T>> T max(T a, T b){

    if (a.compareTo(b) > 0)
        return a;
    else
    return b;
}
```

6.

```java
public class MyType <T , S > {

```

7.

```java
public class MyType <T extends Number, S extends String> {


}
```