Quiz 10

Due Nov 11 at 12pm **Points** 10 **Questions** 10 **Time Limit** None

Instructions

There is no time limit, but you may only make one submission.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	78 minutes	9 out of 10

Score for this quiz: **9** out of 10
Submitted Nov 3 at 6:20pm
This attempt took 78 minutes.

Question 1

1 / 1 pts

Which of the following is NOT a typical advantage of generic methods?

- No need for casting of specified return types
- Faster execution during run-time

Correct!

- Backward-compatibility with older non-generic code
- Compile-time enforcement of specified argument types

Question 2

1 / 1 pts

Consider the following code snippet:

```
ArrayList<Double> arr = new ArrayList<>();
 . . .
```

```
String element = arr.get(0);
```

Is there an error in this code?

- Yes, a compile-time error will occur because you cannot create an `ArrayList` of type `Double`.

Correct!

- Yes, a compile-time error will occur because you cannot assign a `Double` value to a `String` variable.
- Yes, a run-time error will occur because a `Double` value cannot be cast to a `String` value.
- No run-time error or compile-time errors will occur.

Question 3

1 / 1 pts

Internally within the JVM, type parameters are replaced by their bounds (or Object, if unbounded) to become ordinary non-generic classes. What is this simplification process called?

- Type casting

Correct!

- Type erasure
- Type conversion
- Type simplification

Question 4

1 / 1 pts

Which of the following statements about generic methods is correct?

A generic method must reside in a generic class.

The generic type parameter of a generic method designates the method's return type.

When calling a generic method, you need to instantiate the type parameters.

Correct!

A generic method must have a generic type parameter.

Question 5

0 / 1 pts

Consider the following code snippet:

```
public class Box<E>
{
    private E data;
    public Box() { . . . }
    public void insert(E value) { . . . }
    public E getData() { . . . }
}
```

What will result from executing the following code?

```
Box<String> box = new Box<>();
 . . .
box.insert("blue Box");
String b = box.getData();
```

run-time error

'ou Answered

compiler error

orrect Answer

no error

compiler warning

Question 6

1 / 1 pts

Where do you specify the type parameters of a generic method?

- Immediately before the parentheses with the parameters
- Immediately before the return type

Correct!

- First, before any modifiers such as "public" or "static"
- Between the return type and the method name

Question 7

1 / 1 pts

Which of the following statements about using generic programming is NOT correct?

- Using type parameters will potentially avoid some class cast exceptions that may occur when using collections without type parameters.
- Type parameters cannot be used with interfaces.

Correct!

- Using type parameters makes generic code easier to read and understand.
- Using type parameters makes generic code safer.

Question 8

1 / 1 pts

Consider the following code snippet:

```
public static <E extends Comparable<E>> E min(ArrayList<E> objects)
```

What can we conclude about the return type of this method?

Correct!

- (•) The return type is a class that implements the `Comparable` interface.
- ( ) The return type is an array list of generic objects.
- ( ) The return type is a subclass of the `ArrayList` class.
- ( ) The return type is a class that extends the `Comparable` interface.

Question 9

1 / 1 pts

What is used as a wildcard indicator that matches any class in a type parameter?

Correct!

- (•) ?
- ( ) *
- ( ) any
- ( ) &

Question 10

1 / 1 pts

Suppose that `Car` is a subclass of `Vehicle`. Which statement is true about `ArrayList<Car>` and `ArrayList<Vehicle>`?

- ( ) Any Vehicle object can be added into an ArrayList<Car>

Correct!

- (•) Any Car object can be added into an ArrayList<Vehicle>
- ( ) An ArrayList<Car> object can be cast to an ArrayList<Vehicle>
- ( ) ArrayList<Car> is a subclass of ArrayList<Vehicle>

Quiz Score: **9** out of 10