```
  // Get a character from the queue.
  char get() throws QueueEmptyException;
}
```

6. Now, compile the updated **IQChar.java** file. Then, compile **QExcDemo.java**. Finally, run **QExcDemo**. You will see the following output.

```
Attempting to store : A - OK
Attempting to store : B - OK
Attempting to store : C - OK
Attempting to store : D - OK
Attempting to store : E - OK
Attempting to store : F - OK
Attempting to store : G - OK
Attempting to store : H - OK
Attempting to store : I - OK
Attempting to store : J - OK
Attempting to store : K
Queue is full. Maximum size is 10

Getting next char: A
Getting next char: B
Getting next char: C
Getting next char: D
Getting next char: E
Getting next char: F
Getting next char: G
Getting next char: H
Getting next char: I
Getting next char: J
Getting next char:
Queue is empty.
```

Module 9 Mastery Check

1. What class is at the top of the exception hierarchy?
2. Briefly explain how to use **try** and **catch**.
3. What is wrong with this fragment?

```
// ...
vals[18] = 10;
catch (ArrayIndexOutOfBoundsException exc) {
  // handle error
}
```

4. What happens if an exception is not caught?
5. What is wrong with this fragment?

```
class A extends Exception { ...

class B extends A { ...

// ...

try {
  // ...
}
catch (A exc) { ... }
catch (B exc) { ... }
```

6. Can an exception caught by an inner **catch** rethrow that exception to an outer **catch**?
7. The **finally** block is the last bit of code executed before your program ends. True or False? Explain your answer.
8. What type of exceptions must be explicitly declared in a **throws** clause of a method?
9. What is wrong with this fragment?

```
class MyClass { // ... }
// ...
throw new MyClass();
```

10. In question 3 of the Mastery Check in Module 6, you created a **Stack** class. Add custom exceptions to your class that report stack full and stack empty conditions.
11. What are the three ways that an exception can be generated?
12. What are the two direct subclasses of **Throwable**?