Question 1

Given:

```java
public class Main {
    public static void main(String[] args) {
        try(BufferedReader in = new BufferedReader(new InputStreamReader(System.in))) {
            System.out.print("Input: ");
            String input = in.readLine();
            System.out.println("Echo: " + input);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

And the command:

java Main Helloworld

What is the result ?

Options:

A- Input: Echo:

B- Input: Helloworld Echo: Helloworld

C- Input:

Then block until any input comes from System.in.

D- Input:

Echo: Helloworld

E- A NullPointerException is thrown at run time.

Answer:

C

Question 2

Question Type: MultipleChoice

Which statement about a functional interface is true?

Options:

A- It must be defined with the public access modifier.

B- It must be annotated with @FunctionalInterface.

C- It is declared with a single abstract method.

D- It is declared with a single default method.

E- It cannot have any private methods and static methods.

Answer:

C

Question 3

Question Type: MultipleChoice

Given:

```
class CustomType<T> {
    public <T> int count(T[] anArray, T element) {
        int count = 0;
        for(T e : anArray) {
            if (e.equals(element)) ++count;
        }
        return count;
    }
}
```

and

```
public class Test extends CustomType {
    public static void main(String[] args) {
        String[] words = ("banana","orange","apple","lemon"};
        Integer[] numbers = {1,2,3,4,5};
        CustomType type = new CustomType();
        CustomType<String> stringType = new CustomType<>();
        System.out.println(stringType.count(words, "apple"));
        System.out.println(type.count(words, "apple"));
        System.out.printin(type.count (numbers, 3));
    }
}
```

What is the result?

Options:

A- A NullPointerException is thrown at run time.

B- The compilation fails.

C- 1
Null
null

D- 1
1
1

E- A ClassCastException is thrown at run time.

Question 4

Question Type: **MultipleChoice**

Given:

```
public class Tester {
    static class Person implements /* line 1 */ {
        private String name;
        Person(String name) { this.name = name; }
        /* line 2 */
    }
    public static void main(String[] args) {
        Person[] people = {new Person("Joe"),
                           new Person("Jane"),
                           new Person("John")};
        Arrays-sort(people);
        for(Person person: people) {
            System.out.println(person.name);
        }
    }
}
```

You want the code to produce this output:

John

Joe

Jane

Which code fragment should be inserted on line 1 and line 2 to produce the output?

Options:

A- Insert Comparator<Person> on line 1.

Insert

```
public int compare(Person p1, Person p2) {
return p1.name.compare(p2.name);
}
```

on line 2.

B- Insert Comparator<Person> on line 1.

Insert

```
public int compareTo(Person person) {
return person.name.compareTo(this.name);
}
```

on line 2.

C- Insert Comparable<Person> on line 1.

Insert

```
public int compare(Person p1, Person p2) {
return p1.name.compare(p2.name);
}
```

on line 2.

D- Insert Comparator<Person> on line 1.

Insert

```
public int compare(Person person) {
return person.name.compare(this.name);
}
```

on line 2.

Question 5

Question Type: **MultipleChoice**

Given:

```
public class Main {
    public static void main(String[] args) {
        try {
            Path path = Paths.get("/u01/work/filestore.txt");
            boolean result = Files.deleteIfExists(path);
            if(result) System.out.println(path + "is deleted.");
            else System.out.println(path + "is not deleted.");
        } catch(IOException e) {
            System.out.println("Exception");
        }
    }
}
```

Assume the file on path does not exist. What is the result?

Options:

A- The compilation fails.

B- /u01/work/filestore.txt is not deleted.

C- Exception

D- /u01/work/filestore.txt is deleted.

Answer:

A

Question 6

Question Type: MultipleChoice

Given:

```
public class MyResource {
    public MyResource () {
    }
    // Resource methods
}
```

You want to use the myResource class in a try-with-resources statement. Which change will accomplish this?

Options:

A- Extend AutoCloseable and override the close method.

B- Implement AutoCloseable and override the autoClose method.

C- Extend AutoCloseable and override the autoClose method.

D- Implement AutoCloseable and override the close method.

Answer:

D

Question 7

Question Type: **MultipleChoice**

Which interface in the java.util.function package will return a void return type?

Options:

A- Supplier

B- Predicate

C- Function

D- Consumer

Answer:

D