Consider below code:

```
//Test.java
public class Test {
    public static void main(String[] args) {
        final int i1 = 1;
        final Integer i2 = 1;//memory will be decied at the runtime becoz it is
wrapper class Object
        final String s1 = ":ONE";

        String str1 = i1 + s1;// Compiler :: 1 + :ONE => 1:ONE
        String str2 = i2 + s1;// Compiler :: i2 + :ONE

        System.out.println(str1 == "1:ONE");//true
        System.out.println(str2 == "1:ONE");//false
    }
}
```

What will be the result of compiling and executing Test class?

A. true
   true

B. true
   false

C. false
   false

D. false
   true

Answer: B

Q>

Consider below code:

```
//Test.java
public class Test {
    public static void main(String[] args) {
        String javaworld = "JavaWorld";//SCP
        String java = "Java";//SCP
        String world = "World";//SCP
        java += world;// JVM => Java+World => java = JavaWorld(heap area)
        System.out.println(java == javaworld);
    }
}
```

What will be the result of compiling and executing Test class?

A. JavaWorld
B. Java
C. World
D. true
E. false

Answer: E

Q>

What will be the result of compiling and executing Test class?

```
public class Test {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder("SpaceStation");
        sb.delete(5, 6).insert(5, " S").toString().toUpperCase();
        System.out.println(sb);
    }
}
```

A. SPACE STATION

B. SPACE SATION
C. Space Station
D. Space Sation

Q>
```
public class Test {
    public static void main(String[] args) {
        String s1 = "OCP";
        String s2 = "ocp";
        System.out.println(/*INSERT*/);
    }
}
```
Which of the following options, if used to replace /*INSERT*/, will compile successfully and on execution will print true on to the console?
Select 2 options.
A. s1.equals(s2)
B. s1.equals(s2.toUpper())
C. s1.equals(s1.toLower())
D. s1.length()==s2.length()
E. s1.equalsIgnoreCase(s2)

Answer: D,E

Q>
Consider below code of Test.java file:
```
public class Test {
    public static void main(String [] args) {
        String text = "ONE ";
        System.out.println(text.concat(text.concat("ELEVEN ")).trim());
    }
}
```
What will be the result of compiling and executing Test class?
A. ONE ELEVEN
B. ONE ONE ELEVEN
C. ONE ELEVEN ONE ELEVEN
D. ONE ELEVEN ONE

Anser: B

Q>
Consider below code of Test.java file:
```
public class Test {
    public static void main(String[] args) {
        String str = "PANIC";
        StringBuilder sb = new StringBuilder("THET");
        System.out.println(str.replace("N", sb)); //Line n1
    }
}
```
What will be the result of compiling and executing Test class?
A. PATHETIC
B. PANIC
C. Line n1 causes compile time error
D. Line n1 cause runtime error.

Answer: PATHETIC

Q>
Consider below code of Test.java file:
```
public class Test {
```

```
    public static void main(String[] args) {
        boolean flag1 = "Java" == "Java".replace('J', 'J'); //Line n1
        boolean flag2 = "Java" == "Java".replace("J", "J"); //Line n2
        System.out.println(flag1 && flag2);
    }
}
```
What will be the result of compiling and executing Test class?
A. Line n1 causes compilation error.
B. Line n2 causes compilation error.
C. true
D. false

Answer: C

Q>
Consider below code fragment:
```
String place = "MISSS";
System.out.println(place.replace("SS", "T"));
```
What is the output?
A. MIST
B. MITS
C. MISSS
D. MIT

Answer: B

Q>
Consider below code of Test.java file:
```
public class Test {
    public static void main(String[] args) {
        String str = "ALASKA";
        System.out.println(str.charAt(str.indexOf("A") + 1));
    }
}
```
What will be the result of compiling and executing Test class?
A. A
B. L
C. S
D. K
E. RuntimeError

Answer: B

Q>
```
public class Test {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder("TOMATO");
        System.out.println(sb.reverse().replace("O", "A")); //Line n1
    }
}
```
What will be the result of compiling and executing Test class?
A. TOMATO
B. TAMATO
C. TAMATA
D. OTAMOT
E. OTAMAT
F. ATAMAT
G. Compilation Error

Answer: G

Q> .
java.lang.String class implements which of the following interfaces?
Serializable
CharSequence
Comparable
All of the above
Answer: All the above

51.
```
String[] strings = "iNeuron_Technology_privatelimited_Known_For_Java".split("_",
3);
    for (String string : strings)
        System.out.println(string);
```

Answer:iNeuron
Technology
privatelimited_Known_For_Java