Question 1

Examine:

```java
class E1 extends Exception { }

class E2 extends RuntimeException { }

public class App {
    public void m1() {
        System.out.println("m1.Accessed.");
        throw new E1();
    }

    public void m2 () {
        System.out.println("m2.Accessed.");
        throw new E2();
    }
    public static void main (String[] args) {
        int level =1;
        App obj = new App();
        if (level <=5 && level >= 3) {
            obj.m1();
        } else {
            obj.m2();
        }
    }
}
```

Which statement is true?

Options:

A- The program prints m1.Accessed.

B- The program fails compile due to the unhandled E1 exception.

C- The program prints m2.Accessed.

D- The program fails to compile due to the unhandled E2 exception.

Answer:

B

Question 2

Question Type: MultipleChoice

Given:

```
class LogFileException extends Exception {}
class AccessViolationException extends RuntimeException {}

1. public class App {
2.    public static void main (String[] args) throws LogFileException {
3.          App obj = new App ();
4.          try {
5.                obj.open();
6.                obj.process();
7.                    //insert code here
8.          }
9.          catch (Exception e) {
10.               System.out,println("Completed.");
11.         }
12.  }
13.  public void process() {
14.         System.out.println("Processed");
15.         throw new LogFileException();
16.  }
17.  public void open () {
18         System.out.println ("Opened.");
19.         throw new AccessViolationException();
20.  }
21. }
```

Which action fixes the compiler error?

Options:

A- At line 17, add throws AccessViolationException

B- At line 13, add throws LogFileException

C- At line 2, replace throws LogFileException with throws AccessViolationException

D- At line 7, insert throw new LogFileException ();

Answer:

D

Question 3

Question Type: MultipleChoice

Given the code fragment:

```
public class Test {
    public static void main(String[] args) {
        int x;
        /* insert code here */
    }
}
```

Which two code fragments inserted at line 10 print ****?

A

```
x = 3;
do {
    System.out.print("*");
    x--;
} while (x >= 0);
```

B

```
x = 0;
do {
    System.out.print("*");
    x++;
} while (x >= 3);
```

C

```
x = 0;
do {
    System.out.print("*");
    ++x;
}while (x > 3);
```

D

```
x = 3;
do {
    System.out.print("*");
    x--;
}while (x != 1);
```

E

```
x = 0;
do {
    System.out.print("*");
} while (x++ < 3);
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

E- Option E

Answer:

E

Question 4

Question Type: **MultipleChoice**

Given the code fragment:

```
public static void main(String[] args) {
    int ans;
    try {
        int num = 10;
        int div = 0;
        ans = num / div;
    } catch (ArithmeticException ae) {
        ans = 0;                                  // line n1
    } catch (Exception e) {
        System.out.println("Invalid calculation");
    }
    System.out.println("Answer = " + ans);    // line n2
}
```

What is the result?

Options:

A- Answer = 0

B- Invalid calculation

C- Compilation fails only at line n1.

D- Compilation fails only at line n2.

E- Compilation fails at line n1 and line2.

Question 5

Given the code fragment:

```
int num[][] = new int[3][1];
for (int i = 0; i < num.length; i++) {
    for (int j = 0; j < num[i].length; j++) {
        num[i][j] = 10;
    }
}
```

Which option represents the state of the num array after successful completion of the outer loop?

A

```
num[0][0]=10
num[0][1]=10
num[0][2]=10
```

B

```
num[0][0]=10
num[1][0]=10
num[2][0]=10
```

C

```
num[0][0]=10
num[0][1]=0
num[0][2]=0
```

D

```
num[0][0]=10
num[0][1]=10
num[0][2]=10
num[0][3]=10
num[1][0]=0
num[1][1]=0
num[1][2]=0
num[1][3]=0
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Question 6

Question Type: MultipleChoice

Given the code fragment:

```
3. public static void main(String[] args) {
4.        int x = 6;
5.        while (isAvailable(x)) {
6.             System.out.print(x);
7.
8.        }
9. }
10.
11. public static boolean isAvailable(int x) {
12.      return --x > 0 ? true :[ false;
13. }
```

Which modification enables the code to print 54321?

Options:

A- Replace line 6 with System.out.print (--x);

B- At line 7, insert x --;

C- Replace line 5 with while (is Available(--x)) {

D- Replace line 12 with return (x > 0) ? false : true;

Answer:

C

Question 7

Given the code fragment:

```java
class Employee {
    private String name;
    private int age;
    private int salary;


    public Employee(String name, int age) {
        setName(name);
        setAge(age);
        setSalary(2000);
    }


    public Employee(String name, int age, int salary) {
        this(name, age);
        setSalary(salary);
    }


    //getter and setter methods for attributes go here

    public void printDetails() {
        System.out.println(name + " : " + age + " : " + salary);
    }
}
```

Test.java:

```java
class Test {
    public static void main(String[] args) {
        Employee e1 = new Employee();
        Employee e2 = new Employee("Jack", 50);
        Employee e3 = new Employee("Chloe", 40, 5000);

        e1.printDetails();
        e2.printDetails();
        e3.printDetails();
    }
}
```

Which is the result?

A Compilation fails in the Employee class.

B
```
null : 0: 0
Jack : 50 : 0
Chloe : 40 : 5000
```

C
```
null : 0 : 0
Jack : 50 : 2000
Chloe : 40 : 5000
```

D Compilation fails in the Test class.

E Both the Employee class and the Test class fail to compile.

Options:

A- Option A

B- Option B

C- Option C

D- Option D

E- Option E

Answer:

D

Question 8

Question Type: MultipleChoice

Given the code fragment:

```
public class App {
    public static void main(String[] args) {
        String str1 = "Java";
        String str2 = new String("java");
        //line n1
        {
            System.out.println("Equal");
        } else {
            System.out.println("Not Equal");
        }
    }
}
```

Which code fragment, when inserted at line n1, enables the App class to print Equal?

```
  A) Str1.toLowerCase();
     if (str1 == str2)

  B) if (str2.equals(str1.toLowerCase()))

  C) Str1.toLowerCase();
     if (str1.equals(str2))

  D) if (str1.toLowerCase() == str2.toLowerCase())
```

Options:

A- Option A

B- Option B

C- Option C

D- Option D

Answer:

B

Question 9

Given the code fragment:

```
int[] array = {1, 2, 3, 4, 5};
```

And given the requirements:

1. Process all the elements of the array in the reverse order of entry.

2. Process all the elements of the array in the order of entry.

3. Process alternating elements of the array in the order of entry.

Which two statements are true? (Choose two.)

Options:

A- Requirements 1, 2, and 3 can be implemented by using the enhanced for loop.

B- Requirements 1, 2, and 3 can be implemented by using the standard for loop.

C- Requirements 2 and 3 CANNOT be implemented by using the standard for loop.

D- Requirement 2 can be implemented by using the enhanced for loop.

E- Requirement 3 CANNOT be implemented by using either the enhanced for loop or the standard for loop.

Answer:

B, C

Question 10

Question Type: MultipleChoice

Given:

```
interface I {
    public void displayI();
}
abstract class C2 implements I {
    public void displayC2() {
        System.out.print("C2");
    }
}
class C1 extends C2   {
    public void displayI() {
        System.out.print("C1");
    }
}
```

And the code fragment:

```
C2 obj1 = new C1();
I obj2 = new C1();

C2 s = (C2) obj2;
I t = obj1;

t.displayI();
s.displayC2();
```

What is the result?

Options:

A- C1C2

B- C1C1

C- Compilation fails.

D- C2C2

Answer:

A