```
import java.io.FileNotFoundException;
import java.io.IOException;
abstract class Super {
    public abstract void m1() throws IOException;
}
class Sub extends Super {
    @Override
    public void m1() throws IOException {
        throw new FileNotFoundException();
    }
}
public class Test {
    public static void main(String[] args) {
        Super s = new Sub();
        try {
            s.m1();
        } catch (FileNotFoundException e) {
            System.out.print("X");//X
        } catch (IOException e) {
            System.out.print("Y");
        } finally {
            System.out.print("Z");//Z
        }
    }
}
```

A. XZ
B. YZ
C. XYZ
D. compilationerror.

Answer: A

5.

```
import java.io.FileNotFoundException;
import java.io.IOException;

abstract class Super {
    public abstract void m1() throws IOException;
}

class Sub extends Super {
    @Override
    public void m1() throws IOException {
        throw new FileNotFoundException();
    }
}

public class Test {
    public static void main(String[] args) {
        Super s = new Sub();
        try {
            s.m1();
        } catch (FileNotFoundException e) {
            System.out.print("M");
        } finally {
            System.out.print("N");
        }
    }
```

```
}
```
A. MN
B. N
C. CompilationError
D. Program ends abruptly

Answer: C

6.
What will be the result of compiling and executing Test class?

```
public class Test {
    private static void m1() {
        System.out.println(1/0);//ArithmeticException: unchecked
    }

    public static void main(String[] args) {
        try {
            m1();
        } finally {
            System.out.println("A");
        }
    }
}
```

A. A is printed on the console and program terminates normally
B. A is printed on the console,stacktrace is printed and then program ends normally
C. A is printed on the console,stacktrace is printed and then program ends abruptly
D. Compilation Error.

Answer: C

Q>
Which of the following keywords is used to manually throw an exception?
A. throw
B. thrown
C. throws
D. catch

answerL A

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

```
public class Test {
    private static void m1() throws Exception {//Exception :: Checked(partially checked)
        throw new Exception();
    }

    public static void main(String[] args) {
        try {
            m1();
        } finally {
            System.out.println("A");
        }
    }
}
```

A. A is printed on the console and program terminates normally

B. A is printed on the console,stacktrace is printed and then program ends normally
C. A is printed on the console,stacktrace is printed and then program ends abruptly
D. Compilation Error.

Answer: D

Q>
Which of the following is a checked Exception?
A. ClassCastException   ====> RunTimeException
B. FileNotFoundException ===> IOException (checkedException)
C. ExceptionInIntializerError ====> RunTimeException
D. RunTimeException======> parent of all uncheckedException

Answer: B

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

```
public class Test {
    private static String s;
    public static void main(String[] args) {
        try {
            System.out.println(s.length());
        } catch(NullPointerException | RuntimeException ex) {
            System.out.println("DONE");
        }
    }
}
```

A. DONE
B. Executes succesfully but no output
C. CompilationError
D. NoOutput

Answer: C

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

```
class Base {
    public void m1() throws NullPointerException {//NullPointerException :
uncheckedException
        System.out.println("Base: m1()");
    }
}
class Derived extends Base {
    public void m1() throws RuntimeException {//RuntimeException:
uncheckedException
        System.out.println("Derived: m1()");
    }
}

public class Test {
    public static void main(String[] args) {
        Base obj = new Derived();
        obj.m1();
    }
}
```

A. Base:m1()
B. Derived:m1()

C. CompilationError in Derived class
D. CompilationError in Base class

Answer: B

Q>
Consider the following interface declaration:

```
public interface I1 {
    void m1() throws java.io.IOException;//IOException : CheckedException
}
```

Which of the following incorrectly implements interface I1?

A.
```
public class c1 implements I1{
      public void m1(){}
     }
```

B.
```
public class c1 implements I1{
      public void m1()throws java.io.FileNotFoundException{}
   }
```

C.
```
public class c1 implements I1{
      public void m1()throws java.io.IOException{}
   }
```

D.
```
public class c1 implements I1{
      public void m1()throws Exception{}//Since it is throwing Exception ,Parent
can't throw IOException, it should throw either Exception or Throwable
   }
```

Answer: D

Q>.
Which of the following are Java Exception classes? Select 3 options.
A.ClassCastException
B. NullException
C. NumberFormatException
D. IllegalArgumentException
E. ArrayIndexException

Answer: A,C,D

Q>
Given Code:

```
import java.io.*;
class ReadTheFile {
    static void print() { //Line 4
        throw new IOException(); //Line 5
    }
}
public class Test {
    public static void main(String[] args) { //Line 10
        ReadTheFile.print(); //Line 11
        //Line 12
    }
}
```

Which 2 changes are necessary so that code compiles successfully?

A. replace Line 4 with static void print() throws Exception
B. replace Line 4 with static void print() throws Throwable

C. replace Line 10 with public static void main(String[] args) throws IOException
D. Surrond Line 11 with
```
try{
 ReadTheFile.print();
}catch(IOException | Exception e){}
```
E. Surrond Line 11 with
```
try{
 ReadTheFile.print();
}catch(Exception e){}
```
F. Surrond Line 11 with
```
try{
 ReadTheFile.print();
}catch(IOException  e){}
```

Answer: A,E

.