Java: Exception

Quiz

Question 1

```
class A {

   public static void main (String[] args) {

   Object error = new Error();

   Object runtimeException = new RuntimeException();

    System.out.print((error instanceof Exception) + ",");


   System.out.print(runtimeException instanceof Exception);
}}
```

Code prints

A.   false,false

B.   false,true

C.   true,false

D.   true,true

Question 2

```
class Test1{
static void display() throws Exception{
System.out.println("Hello");}
public static void main(String[] args) {
display();
}}
```

Which of the following are possible causes of compilation error in the code?

A. `display()` does not throw any Exception .

B. `display()` is incorrectly declared

C. `main()` must provide `a try-catch` block for `display()`

D. `main()` must be declared with `throws Exception` clause .

Question 3

Assume the following method declaration:

```
protected void f()throws java.io.IOException{}
```

Which of the following is NOT right way to override f()?

Given that **FileNotFoundException is subclass of IOException**

A. `protected void f()throws Exception`

B. `public void f() throws FileNotFoundException{}`

C. `public void f()`

D. `public void f() throws Error`

Question 4

```
class A {A() throws Exception {}} // line 1
class B extends A {B() throws Exception {}} //
 line  2
class C extends A {C(){}} // line 3
```

What will happen on compilation of the code?

A. Compilation error at line 1

B. Compilation error at line 2

C. Compilation error at line 3

D. Code compiles clean

Question 5

Which of the following are checked exceptions?

A. **ArrayIndexOutOfBoundsException**

B. **IllegalArgumentException**

C. **CloneNotSupportedException**

D. **NullPointerException**

Question 6

Which of the following are unchecked exceptions?

A. **IOException**

B. **IllegalArgumentException**

C. **SQLException**

D. **AssertionError**

```
class A{
public static void main(String[] args) {
try{
try{
int i=10/0;
}catch( Exception e){
        System.out.println("caught inside");}
}catch(ArithmeticException e){
    System.out.println("caught outside");
}}}
```

What is the result of compilation/execution of the code?

```
A.   Code does not compile

B.   Code compiles and prints nothing

C.   Code compiles and prints "caught outside"

D.   Code compiles and prints "caught inside"
```

Question 8

```java
public static void main(String[] args) {
try{
int i=10/0;
}catch( Exception e){
        System.out.println("caught inside");}
catch(ArithmeticException e){
    System.out.println("caught outside");
}}}
```

What is the result of compilation/execution of the code?

A. Code does not compile

B. Code compiles and prints nothing

C. Code compiles and prints `"caught outside"`

D. Code compiles and prints `"caught inside"`

Question 9

A **try** block

A. Should have at least 1 **catch** block

B. Should have at least 1 **finally** block

C. Should have either a **catch** block or **finally** block

D. Should have both a **catch** block and **finally** block

Question 10

```
class Test1{
public static void m1(int i) {
try{
if(i<0) return;
int j=10/i;
}catch( Exception e){System.out.println("caught ");}
finally{System.out.println("thanks");}
System.out.println("bye");
}
public static void main(String[] args) {
m1(-10);}}
```

A code prints which of the following string(s)?

A.   caught

B.   thanks

C.   bye

D.   None of them

Question 12

Assume the following method declaration:

```
protected void f() throws
  CloneNotSupportedException
```

Which of the following is NOT the right way to override f()?

A. `protected void f()throws Exception`

B. `public void f()throws RuntimeException`

C. `public void f()`

D. `public void f() throws Error`

Question 13

```
public class Flower implements Cloneable{
// insert method declaration here
{(Flower) super.clone();}
}
```

Which of the following is/are NOT right declaration for clone method?

A. `public Object clone()`

B. `protected Object clone()throws CloneNotSupportedException`

C. `Object clone()throws CloneNotSupportedException`

D. `public Flower clone()`