

Which of the following is the parent class of all the Exception classes?

Options:

- ☐ Exception
- ☐ ClassNotFoundException
- ☐ Error
- ☒ Throwable

✓ Correct Answer

When does an exception arise in a program?

Options:

- ☐ At compile time
- ☒ At runtime
- ☐ At any time
- ☐ None of the above.

✓ Correct Answer

Which of the following keywords is not related to exception handling?

Options:

- ☐ finally
- ☐ catch
- ☐ throw
- ☒ thrown

✔ Correct Answer

In which of the following blocks which write the code that might throw an exception?

Options:

- ☐ catch
- ☐ finally
- ☒ try
- ☐ None of the above

✔ Correct Answer

Which of these keywords is used to explicitly throw an exception?

Options:

- ☐ throws
- ☒ throw
- ☐ try
- ☐ catch

✓ Correct Answer

Which exception is thrown when a string contains null?

Options:

- ☐ ArithmeticException
- ☐ ClassNotFoundException
- ☒ NullPointerException
- ☐ None of the above


✓ Correct Answer

What will be the output of the following code?

```
public class Test {  
  
    public static void main(String args[]) {  
  
        try {  
            throw 10;  
        }  
        catch(int e) {  
            System.out.println("Catch the exception" + e);  
        }  
    }  
}
```

Options:

- ☐ Catch the exception 10;
- ☐ Catch the exception 0;
- ☒ Compilation error
- ☐ Runtime error


 Correct Answer

What will be the output of the following code?

```
public class Test {  
  
    public static void main(String args[]) {  
  
        int a = 10;  
        int b = 0;  
        int c = a / b;  
        System.out.println(c);  
    }  
}
```

Options:

- ☐ Compile fine
- ☐ Compile and run fine
- ☐ Compilation error
- ☒ Compile fine but throws ArithmeticException

 Correct Answer

What will be the output of the following code?


```
public class Test
{
    public static void main (String[] args)
    {
        try
        {
            int a = 0;
            System.out.print ("a = " + a + " ");
            int b = 20 / a;
            System.out.print ("b = " + b);
        }

        catch(ArithmeticException e)
        {
            System.out.print ("Divide by zero error ");
        }

        finally
        {
            System.out.print ("Inside the finally block");
        }
    }
}
```

Options:

- ☐ Compile error
- ☐ Divide by zero error
- ☐ Inside the finally block
- ☒ a = 0 Divide by zero error Inside the finally block

 Correct Answer

What will be the output of the following code?

```
class Test extends Exception { }

public class Main {

    public static void main(String args[]) {


        try {
            throw new Test();
        }

        catch(Test t) {
            System.out.print("Catch the Test Exception,");
        }

        finally {
            System.out.print("Inside finally block ");
        }
    }
}
```

Options:

- ☐ Catch the Test Exception,
- ☐ Inside finally block
- ☐ Runtime error
- ☒ Catch the Test Exception, Inside the finally block

 Correct Answer