1——

(a) throwable

(b) Exceptions (members of the Exception family) are thrown to signal abnormal conditions that can often be handled by some catcher, such as AssertionError when we try to use some method but we use the arguments that not satisfied the precondition. But exceptions are possible to not be caught and therefore could result in a dead thread.

Errors are usually thrown for more serious problems, such as out of memory, that may not be so easy to handle.

In general, code you write should only throw exceptions but not error.

(c) Exceptions provide the mean to separate the details of what to do when something out of ordinary happens from the main logic of a program. With exceptions, we can do more about the error detection, reporting and handling.

Here is an example. In A1, when we try to use method setAdvisor1(), the argument is not allowed to be null. So we can use the try/catch to see if there is any error when we use this method if we use the argument that not meet the precondition. If it doesn't meet the precondition, the program would not fail but give a message about the error.

2——

```
Throwable
        Exceptions
                ActivationException
                ApplicationException
                RuntimeException
                        AnnotationTypeMismatchException
                        ArithmeticException
        Error
                AnnotationFormatError
                AssertionError
```

3——

(1) if S1 throws an exception: Both S1 and S3 would not be executed
(2) if S1 does not throw an exception: Both S1 and S3 will be executed

4——

throw new ArithmeticException("args should not be negative");

5——

(a)

```
six
four
three
two
Exception in thread "main" java.lang.ArithemeticException: / by zero
```

(b)

```
six
five
four
three
two
one
```

(c)

```
six
five
```

6——

if(m == n) throw new RuntimeException("min of 0 values doesn't exist");

7———

```
try{
	int min = c[0];
	for (int k=0; k < c.length; k = k +1){
		if (c[k] < min) min = c[k];
	}
	return min;
}
catch (ArrayIndexOutOfBoundsException e){
	throw new RuntimeException("min of 0 values doesn't exist");
}
```

8———

if k != 0 and k!= 6, then would print out:

```
0
3
4
5
7
1
```

if k = 0, then would print out:

```
0
3
6
7
1
```

if k = 6, then would print out:

```
0
3
4
2
```

9———

```
public class A{
	public static void main(String[] args){
		/**Print the sum of two integers read from the keyboard*/
		System.out.println("enter a number");
		String s = "a12";
		int a = 0;
		int b = 0;
		try{
			a = Integer.parseInt(s);
			throw new NumberFormatException();
		}catch (NumberFormarException e){
			a = 1;
		}
		System.out.println("Enter another number:");
		try{
			b = Integer.parseInt(s);
			throw new NumberFormatException();
		}catch (NumberFormarException e){
			b = 1;
		}
		System.out.println("Product:" + a*b);

	}
}
```