Started on	Tuesday, 6 September 2022, 6:35 PM
State	Finished
Completed on	Tuesday, 6 September 2022, 6:36 PM
Time taken	50 secs
Marks	7.00/7.00
Grade	**10.00** out of 10.00 (**100**%)

Question **1**

Correct

Mark 1.00 out of 1.00

A switch statement, most often has the form:
switch (expression) {
case constant-1:
statements-1
break;
...
}

The value of the expression can be:
i. int
ii. short
iii. byte
iv. Primitive char
v. Enum
vi. String
vii. Real number

Select one:

- a. iii , iv and v
- b. i , ii, iii and iv
- c. All, except vii ✔
- d. vi and vii
- e. All of the types listed

The correct answer is: All, except vii

Question **2**

Correct

Mark 1.00 out of 1.00

The following code writes out the name of a day of the week depending on the value of *day*. True or False?

```
String dayName = null;
switch (day) {
case 1:
dayName = "Sunday";
break;
case 2:
dayName = "Monday";
break;
case 3:
dayName = "Tuesday";
break;
case 4:
dayName = "Wednesday";
break;
case 5:
dayName = "Thursday";
break;
case 6:
dayName = "Friday";
break;
case 7:
dayName = "Saturday";
break;
}
System.out.println(dayName);
```

Select one:

- True ✔
- False

The correct answer is 'True'.

Question **3**

Correct

Mark 1.00 out of 1.00

Given the following piece of code:

```
class CostCalculationException extends Exception{}
class Item {
      public void calculateCost() throws CostCalculationException {
            //...
            throw new CostCalculationException();
            //...
      }
}
class Company {
       public void payCost(){
            new Item().calculateCost();
      }
}
```

Which of the following statements is correct?

Select one or more:

- a. This code will compile without any problems.
- b. This code will compile if in method payCost()you return a boolean instead of void.
- c. This code will compile if you add a try-catch block in payCost() ✔
- d. This code will compile if you add throws CostCalculationException in the signature of method payCost(). ✔

The correct answers are: This code will compile if you add a try-catch block in payCost(), This code will compile if you add throws CostCalculationException in the signature of method payCost().

Question **4**

Correct

Mark 1.00 out of 1.00

Given the following piece of code:

```
class Student { public void talk(){} }
public class Test{
        public static void main(String args[]){
                Student t = null;
                try {
                    t.talk();
                    } catch(NullPointerException e){
                            System.out.print("There is a NullPointerException. ");
                    } catch(Exception e){
                            System.out.print("There is an Exception. ");
                    }
                    System.out.print("Everything ran fine. ");
        }
}
```

what will be the result?

a. If you run this program, the following is printed:
There is a NullPointerException. Everything ran fine.

b. If you run this program, the following is printed:
There is a NullPointerException.

c. If you run this program, the following is printed:
There is a NullPointerException. There is an Exception.

d. This code will not compile, because in Java there are no pointers.

Select one:

- a. ✔
- b.
- c.
- d.

The correct answer is: a.

Question **5**

Correct

Mark 1.00 out of 1.00

Consider the following code (assume that comments are replaced with real code that works as specified):

```
public class TestExceptions {

  static void e() {
    // Might cause any of the following unchecked exceptions to be
    // thrown:
    // Ex1, Ex2, Ex3, Ex4
  }

  static void April() {
    try {
      e();
    } catch (Ex1 ex) {
      System.out.println("April caught Ex1");
    }
  }

  static void March() {
    try {
      April();
    } catch (Ex2 ex) {
      System.out.println("March caught Ex2");
      // now cause exception Ex1 to be thrown
    }
  }

  static void February() {
    try {
      March();
    } catch (Ex1 ex) {
      System.out.println("February caught Ex1");
    } catch (Ex3 ex) {
      System.out.println("February caught Ex3");
    }
  }

  static void January() {
    try {
      February();
    } catch (Ex4 ex) {
      System.out.println("January caught Ex4");
      // now cause exception Ex1 to be thrown
    } catch (Ex1 ex) {
      System.out.println("January caught Ex1");
    }
  }

  public static void main(String[] args) {
     January();
  }
```

}

Assume now that this program is run four times. The first time, method e throws exception Ex1, the second time, it throws exception Ex2, etc.

What are the results of the four runs (a or b)?

a.

1. The program prints:
   April caught Ex1
2. The program prints:
   March caught Ex2
   February caught Ex1
3. The program prints:
   February caught Ex3
4. The program prints:
   January caught Ex4
   And execution stops due to an uncaught exception Ex1 thrown in main()

b.

1. The program prints:
   April caught Ex3
2. The program prints:
   March caught Ex2
   February caught Ex2
3. The program prints:
   March caught Ex3
4. The program prints:
   January caught Ex4
   And execution stops due to an uncaught exception Ex1 thrown in main()

Select one:

- (•) a. ✔
- ( ) b.

The correct answer is: a.

◄ Learning Journal Unit 1

Jump to...

Correct

Mark 1.00 out of 1.00

Lab 1 Unit 1 ►

Which statements are correct regarding Java's predefined class called *Throwable*?

Select one or more:

- [x] a. The class Throwable represents all possible objects that can be thrown by a throw statement and caught by a catch clause in a try...catch statement. ✔
- [x] b. The thrown object must belong to the class Throwable or to one of its (many) subclasses such as Exception and RuntimeException. ✔
- [x] c. The object carries information about an exception from the point where the exception occurs to the point where it is caught and handled. ✔
- [x] d. A Throwable contains a snapshot of the execution stack of its thread at the time it was created. ✔

The correct answers are: The class Throwable represents all possible objects that can be thrown by a throw statement and caught by a catch clause in a try...catch statement., The thrown object must belong to the class Throwable or to one of its (many) subclasses such as Exception and RuntimeException., The object carries information about an exception from the point where the exception occurs to the point where it is caught and handled., A Throwable contains a snapshot of the execution stack of its thread at the time it was created.

Question **7**

Correct

Mark 1.00 out of 1.00

"Subclasses of the class *Exception* which are not subclasses of RuntimeException require mandatory exception handling." What are the practical implications of this statement?

Select one or more:

- [x] a. If a method can throw such an exception, then it must declare this fact by adding a throws clause to the method heading. ✔
- [x] b. If a routine includes any code that can generate such an exception, then the routine must deal with the exception. ✔
- [ ] c. The routine cannot handle the exception by adding a throws clause to the method definition.
- [x] d. The routine can handle the exception by including the code in a try statement that has a catch clause to handle the exception. ✔

The correct answers are: If a method can throw such an exception, then it must declare this fact by adding a throws clause to the method heading., If a routine includes any code that can generate such an exception, then the routine must deal with the exception., The routine can handle the exception by including the code in a try statement that has a catch clause to handle the exception.