Guidance and Sample questions for the final exam:

This preparation document is considered as part of the content that is required for the preparation for the final exam.

The following questions are provided as sample questions covering only a part of the required content for the exam. The exam covers other required topics including mainly and not limited to:

Java Memory management, Inheritance, Constructors, type casting, Graphical User Interface, Abstract Data Types, Exception handling, Recursion, Streams and files.

[1] Which of the following statements is true ?

a) Every Java class must contain a constructor implementation.
b) Every Java class must override equals(Object o) method.
c) Every Java class must override toString(Object o) method.
d) Statements a), b), and c) are all correct.
e) None of statements a), b), and c) are correct.

[2] Which of the following statements is true ?

a) A class may not implement an abstract method inherited from an abstract class.
b) A class may not implement some methods of its implemented interface.
c) The java instanceof operator is used to test whether the object is an instance of the specified type (class or subclass or interface).
d) A class inherits its superclass constructor.
e) Statements a), b) , c) and d) are all correct.

f) None of statements a), b), and c) are correct.

[3] We consider an instance obj of a certain class named TestClass; we consider the following code :

```
Systeme.out.println( obj.getClass().getName());
```

When we execute this code it will display:

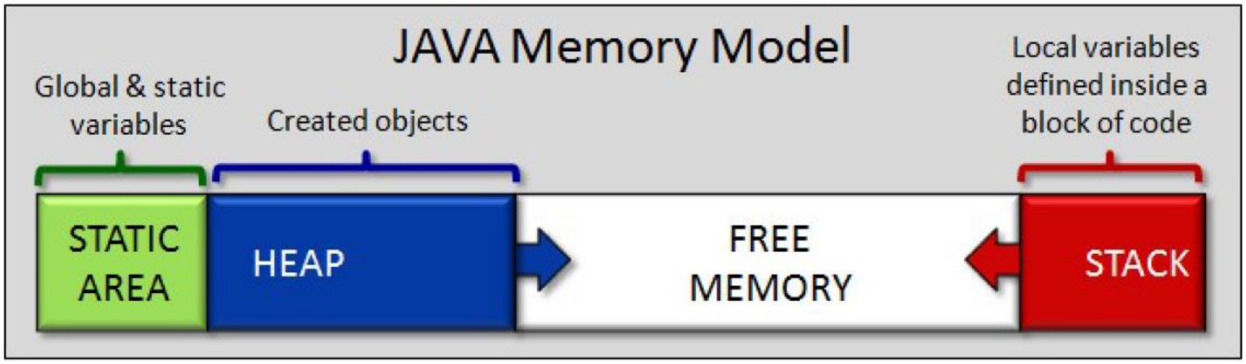
a) TestClass
b) obj is instanceof TestClass
c) An error message
d) Non of the above

[4] Consider the following code declaring and setting variables:

```
int i = 10; int[] a = new int[10];     String s= "abc";
String[] sa = { "abc", "abcd" };
Object myObj = new Object();
```

Which of the following statement is correct:

a) s, a, myObj and sa refer to objects
b) s is of a primitive
c) myObj, i and s are of primitive types
d) Only myObj and sa are not of primitive type
e) non of the above is true.


JAVA Memory Model
Global & static variables
Created objects
Local variables defined inside a block of code
STATIC AREA
HEAP
FREE MEMORY
STACK

[5] Consider the following numbered statement about memory storage:

1.The Static Area of memory is memory that is used by the global & static variables that are defined by your program. This memory usage is fixed and does not change as the program runs.
2. The Free Memory is the memory that is not currently being used by your program. If this memory ever gets used up during your program, you will get an "Out Of Memory" error and your program will stop running.
3. The Stack memory is the memory that is used to store local variables. It also gets used up a little each time you call a method or run code within a block of code (i.e., a block is any code within braces). The amount of memory used during a method call depends on the number and size of the local variables defined in the method as well as its parameters.
4. The Heap memory is the memory that stores all the objects that you create. Each time that you call a constructor by using the new keyword, the Heap memory will increase.

Which of the following statement is True:

a) only statement 1 is true
b) only statement 2 is true
c) only statement 3 is true
d) only statement 4 is true

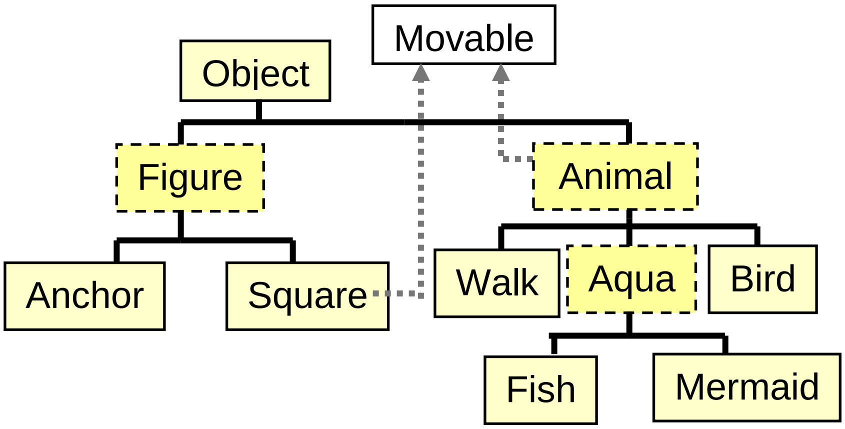
e) all four statements are true

The solution of this question is left as an exercise to the reader who is encouraged to consult chapter 2 to find the correct answer.

Consider the following class hierarchy which defines:

- 3 *abstract* classes (**Figure**, **Animal**, **Aqua**)
- 6 *concrete* classes (**Anchor**, **Square**, **Walk**, **Fish**, **Mermaid**, **Bird**)
- 1 *interface* (**Movable**)

Assume that each class has a zero-parameter constructor.



[6] Choose the right line of code that check if an object x is an instance of Mermaid

```
a) x instanceof Mermaid
```

b) x == **Mermaid**

c) x.getClass().equals("**Mermaid** ")

d) x.getClass() == **Mermaid**

e) x.getClassName().equals("**Mermaid** ");

[7] Which one of the following lines of code **will** compile:

a) Animal v = **new** Animal();

b) Anchor e = (Figure)**new** Anchor();

c) Square c = **new** Anchor();

d) Object x = **new** Bird();

e) Movable i = **new** Anchor();

[8] Which one of these pieces of code results in **true** ?

a) **new** Figure() **instanceof** Square

b) **new** Fish() **instanceof** Movable

c) (**new** Bird()).getClass() == "Bird"

d) Mermaid **instanceof** Animal

e) (**new** Movable()).getClass() == Movable

[9] which of the following operation may result in a **InputMismatchException** exception:

a)Writing to a file

b) Reading in a file

c)reading an integer with a instance of Scanner:

```
number1 = keyboard.nextInt();
```

e) Non of the above.

[10] When an exception is thrown in a try block which is handled with three catch blocks, indicate the block that will be executed to handle the exception:

a) The first block is executed
b) Only the first block matching the type of exception is executed, then if there is a finally block it will be executed;
c) All the catch block are executed according to their order of specification;
d) None of the above.

[11] which block of the following code will never be executed:

```
public void getCustomerInfo() {
 /* a)*/ try {
//  something that may cause an exception: block a
}
/*  b) */ catch (Exception ex){
// Catche block b
}
/* C)*/ catch (java.io.IOException ex){
// block c
}
   /* d)*/ catch (java.io.FileNotFoundException ex){
// block d
}
}
```

a) Block a;
b) block b;
c) block c and block d
d) block b, c and d

[12] Consider the following code:

```
import java.io.*;
public class FileOutputStreamTestProgram {
public static void main(String[] args) {
try {
FileOutputStream out;
/*block 1 */ out = new FileOutputStream("myFile.dat");
/*block 2 beginning */ out.write('H'); out.write(69);
out.write(76); out.write('L');
out.write('O'); out.write('!'); /* end of block 2*/
 out.close();
}
/*block 3 */  catch (FileNotFoundException e) {
System.out.println("Error: Cannot open file for writing");
}
 /* block 4*/ catch (IOException e) {
System.out.println("Error: Cannot write to file");
}
}
}
```

If you have to create two try blocks splitting the existing try block in the indicated block1 and block2, choose from the below options the correct associations relating try block1 and block2 to catch blocks:

a) (Block1 ,block3) and (block 2, block 4)
b) (Block1 ,block3 and block4) and (block 2, block 4)
c) (Block1 ,block3) and (block 2,block3 and block 4)
d) Non of the above

[13] To detect the event of reaching the end of a file we use:

a) **The method of a** FileInputStream available() that counts the remaining not yet read bytes; or **The method ready() of BefferedReader that returns false when no more lines to read.**
b) **Both of the above since the input stream is wrapped by the BefferedReader.**
c) **Non of the above.**