Tasks till 02/06/2018  
  
1. Write a short note on Deque and give example code.

Deque is a queue in which we can add or remove elements from both sides.

import java.util.Deque;

import java.util.ArrayDeque;

public class Dequeexamole {

public static void main(String[] args) {

Deque<String> dq = new ArrayDeque<String>();

dq.add("Glenn");

dq.add("Negan");

dq.addLast("Maggie");

dq.addFirst("Rick");

dq.add("Daryl");

System.*out*.println("Elements in Deque:"+dq);

System.*out*.println("Removed element: "+dq.removeLast());

System.*out*.println("Head: "+dq.element());

System.*out*.println("poll(): "+dq.pollLast());

System.*out*.println("peek(): "+dq.peek());

System.*out*.println("Elements in Deque:"+dq);

}

}  
2. Write a short note on Generics an all types of Parameters used in Generics with example code.

class Test<T>

{

    T obj;

    Test(T obj) {  this.obj = obj;  }  // constructor

    public T getObject()  { return this.obj; }

}

class Main

{

    public static void main (String[] args)

    {

        Test <Integer> iObj = new Test<Integer>(67);

        System.out.println(iObj.getObject());

        // instance of String type

        Test <String> sObj =

                          new Test<String>("Generics Example");

        System.out.println(sObj.getObject());

    }

}

3. Write a short note on Map Interface.

A map contains values on the basis of key i.e. key and value pair. Each key and value pair is known as an entry. Map contains only unique keys. Map is useful if you have to search, update or delete elements on the basis of key.

4. Write the difference between LinkedList and ArrayList.

ArrayList internally uses dynamic array to store the elements whereas LinkedList is internally uses doubly linked list to store the elements. Manipulation with ArrayList is slow because it internally uses array. Manipulation with LinkedList is faster than ArrayList because it uses doubly linked list so no bit shifting is required in memory. ArrayList class can act as a list only because it implements List only. LinkedList class can act as a list and queue both because it implements List and Deque interfaces.

5. Write a note on Dynamic array in java.

6. Write the difference between checked and unchecked exception with example code

In Java exceptions under Error and RuntimeException classes are unchecked exceptions, everything else under throwable is checked. Consider the following Java program. It compiles fine, but it throws ArithmeticException when run. The compiler allows it to compile, because ArithmeticException is an unchecked exception.

-Checked Exception

import java.io.\*;

class Main {

    public static void main(String[] args) throws IOException {

        FileReader file = new FileReader("C:\\test\\a.txt");

        BufferedReader fileInput = new BufferedReader(file);

        // Print first 3 lines of file "C:\test\a.txt"

        for (int counter = 0; counter < 3; counter++)

            System.out.println(fileInput.readLine());

        fileInput.close();

    }

}

-Unchecked Exception

class Main {

   public static void main(String args[]) {

      int x = 0;

      int y = 10;

      int z = y/x;

  }

}

7. Write the difference between throw and throws with example code

Throws keyword is used to declare an exception. Basic Difference:

1.throw is used to explicitly throw a exception, but throws is used to mention all the exceptions which may be thrown by that function as an alert to the calling function. 2.throws is used to in function definitions, throw is used inside a function.

-Throw

public class Example1{

void checkAge(int age){

if(age<18)

throw new ArithmeticException("Not Eligible for voting");

else

System.out.println("Eligible for voting");

}

public static void main(String args[]){

Example1 obj = new Example1();

obj.checkAge(13);

System.out.println("End Of Program");

}

}

-Throws

public class Example1{

int division(int a, int b) throws ArithmeticException{

int t = a/b;

return t;

}

public static void main(String args[]){

Example1 obj = new Example1();

try{

System.out.println(obj.division(15,0));

}

catch(ArithmeticException e){

System.out.println("You shouldn't divide number by zero");

}

}

}

8. Write a note or nested try…catch block with example code

When a [try catch block](https://beginnersbook.com/2013/04/try-catch-in-java/) is present in another try block then it is called the nested try catch block. Each time a try block does not have a catch handler for a particular [exception](https://beginnersbook.com/2013/04/java-exception-handling/), then the catch blocks of parent try block are inspected for that exception, if match is found that that catch block executes.If neither catch block nor parent catch block handles exception then the system generated message would be shown for the exception, similar to what we see when we don’t handle exception.

class Nest{

public static void main(String args[]){

try{

try{

System.out.println("Inside block1");

int b =45/0;

System.out.println(b);

}

catch(ArithmeticException e1){

System.out.println("Exception: e1");

}

try{

System.out.println("Inside block2");

int b =45/0;

System.out.println(b);

}

catch(ArrayIndexOutOfBoundsException e2){

System.out.println("Exception: e2");

}

System.out.println("Just other statement");

}

catch(ArithmeticException e3){

System.out.println("Arithmetic Exception");

System.out.println("Inside parent try catch block");

}

catch(ArrayIndexOutOfBoundsException e4){

System.out.println("ArrayIndexOutOfBoundsException");

System.out.println("Inside parent try catch block");

}

catch(Exception e5){

System.out.println("Exception");

System.out.println("Inside parent try catch block");

}

System.out.println("Next statement..");

}

}

9. Write a note on MultiThreading and MultiTasking

Multithreading in java is a process of executing multiple threads simultaneously.Thread is basically a lightweight sub-process, a smallest unit of processing. Multiprocessing and multithreading, both are used to achieve multitasking.But we use multithreading than multiprocessing because threads share a common memory area. They don't allocate separate memory area so saves memory, and context-switching between the threads takes less time than process.Java Multithreading is mostly used in games, animation etc.It doesn't block the user because threads are independent and you can perform multiple operations at same time.You can perform many operations together so it saves time. Threads are independent so it doesn't affect other threads if exception occur in a single thread.

Multitasking is a process of executing multiple tasks simultaneously. We use multitasking to utilize the CPU. Multitasking can be achieved by two ways:

Process-based Multitasking

Thread-based Multitasking

10. Explain regular expression and quantifiers with example code

A regex quantifier tells the regex engine to match a character or group of characters for specified number of times . Quantifiers allow user to specify the number of occurrences to match against.

11. Explain DAO design pattern with example

Data Access Object or DAO design pattern is a popular design pattern to implement persistence layer of Java application. DAO pattern is based on [abstraction](http://javarevisited.blogspot.sg/2010/10/abstraction-in-java.html) and [encapsulation](http://javarevisited.blogspot.sg/2012/03/what-is-encapsulation-in-java-and-oops.html) design principles and shields rest of application from any change in the persistence layer e.g. change of database from Oracle to MySQL, change of persistence technology e.g. from File System to Database. For example, if you are authenticating the user using a relational database and later your company wants to use [LDAP to perform authentication](http://javarevisited.blogspot.sg/2011/11/ldap-authentication-active-directory.html). If you are using DAO design pattern to access database, it would be relatively safe as you only need to make a change on Data Access Layer. DAO design pattern also keeps coupling low between different parts of an application. By using DAO design pattern your View Layer is completely independent of DAO layer and only Service layer has the dependency on it which is also abstracted by using DAO interface.

12. Write a note on how DAO helps in abstraction

13. Describe the different approaches of String processing.

14. What is the difference between System.out ,System.err and System.in?

15. What is the purpose of the System class?

16. Which is the abstract parent class of FileWriter ?

17. Which class is used to read streams of characters from a file?

18. Which class is used to read streams of raw bytes from a file?

19. What are the differences between FileInputStream/FileOutputStream and RandomAccessFile

20. Write a note on Channels and Buffer with example.