Assignment 4

1. What's the difference between final, finally? What is finalize()?

   final is the keyword and access modifier which is used to apply restrictions on a class, method, or variable. finally is the block in Java Exception Handling to execute the important code whether the exception occurs or not. finalize is the method in Java which is used to perform clean up processing just before object is garbage collected.

2. What's the difference between throw and throws?

   The throw keyword is used inside a function. It is used when it is required to throw an Exception logically. The throws keyword is used in the function signature. It is used when the function has some statements that can lead to exceptions.

3. What are the two types of exceptions?

   Check and unchecked exception.

4. What is error in java?

   An Error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch.

5. Exception is object, true or false?

   True. An exception object is an instance of an exception class. It gets created and handed to the Java runtime when an exceptional event occurred that disrupted the normal flow of the application.

6. Can a finally block exist with a try block but without a catch?

   Yes, finally block always executes even if you have exception or return statement in try block.

7. From java 1.7, give an example of the try-resource feature.

   ```
   static String readFirstLineFromFile(String path) throws IOException {
     try (BufferedReader br =
                   new BufferedReader(new FileReader(path))) {
       return br.readLine();
   ```

```
    }
}
```

8. What will happen to the Exception object after exception handling?

   The Exception object will be garbage collected in the next garbage collection.

9. Can we use String as a condition in switch(str){} clause?

   Yes.

10. What's the difference between ArrayList, LinkedList and vector?

ArrayList	LinkedList
This class uses a dynamic array to store the elements in it.	This class uses a doubly linked list to store the elements in it.
This class implements a List interface. Therefore, this acts as a list.	This class implements both the List interface and the Deque interface. Therefore, it can act as a list and a deque.
This class works better when the application demands storing the data and accessing it.	This class works better when the application demands manipulation of the stored data.

   Vector is thread-safe.

11. What's the difference between hashTable and hashMap?

    HashMap is non-synchronized whereas Hashtable is synchronized.
    HashMap allows one null key, which will be in the first bucket(index 0), while don't allow null key.

12. What is static import?

    With the help of static import, we can access the static members of a class directly without class name or any object.

13. What is static block?

The static block is a block of statement inside a Java class that will be executed when a class is first loaded into the JVM. A static block helps to initialize the static data members, just like constructors help to initialize instance members.

14. Explain the keywords:

default(java 1.8), break, continue, synchronized, strictfp, transient, volatile, instanceOf

default allows the interfaces to have methods with implementation without affecting the classes that implement the interface.

The break statement in java is used to terminate from the loop immediately.

The continue statement in Java is used to skip the current iteration of a loop.

synchronized is only one thread can access the resource at a given point in time.

strictfp is used in java for restricting floating-point calculations and ensuring the same result on every platform while performing operations in the floating-point variable.

transient is a variables modifier used in serialization. At the time of serialization, if we don't want to save value of a particular variable in a file, then we use transient keyword. When JVM comes across transient keyword, it ignores original value of the variable and save default value of that variable data type.

Volatile keyword is used to modify the value of a variable by different threads. It is also used to make classes thread safe. It means that multiple threads can use a method and instance of the classes at the same time without any problem, any write to a volatile field happens before every subsequent read of the same field.

instanceof is a binary operator used to test if an object is of a given type.

15. Create a program including two threads – thread read and thread write.

Input file ->Thread read -> Calculate -> buffered area Buffered area -> Thread write -> output file
Detailed description is in assignment4.txt file. Sample input.txt file.

Attached files are input.txt and a more detailed description file.

```
public class Assignment4 {
    public static void main(String[] args) {
        BlockingQueue<String> queue = new ArrayBlockingQueue<String>(1024);
```

```
        ReaderThread reader = new ReaderThread(queue);
        WriterThread writer = new WriterThread(queue);

        new Thread(reader).start();
        new Thread(writer).start();
    }
}

class ReaderThread implements Runnable {

    protected BlockingQueue<String> blockingQueue = null;

    public ReaderThread(BlockingQueue<String> blockingQueue) {
        this.blockingQueue = blockingQueue;
    }

    @Override
    public void run() {
        BufferedReader br = null;
        try {
            br = new BufferedReader(new FileReader(new File("input.txt")));
            String buffer = null;
            while ((buffer = br.readLine()) != null ) {
                if (buffer.equals("")) continue;
                String[] s = buffer.split("\\s");
                int res = Integer.parseInt(s[0]);
                for(int i = 1; i < s.length; i++){
                    if(s[i].equals("+")){
                        res += Integer.parseInt(s[i+1]);
                    }
                    if(s[i].equals("-")){
                        res -= Integer.parseInt(s[i+1]);
                    }
                }
                String val = buffer + " = " + res;
                blockingQueue.put(val);
            }
            blockingQueue.put("EOF");  //When end of file has been reached

        } catch (FileNotFoundException e) {

            e.printStackTrace();
        } catch (IOException e) {

            e.printStackTrace();
        } catch (InterruptedException e) {

        } finally {
            try {
                br.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
```

```
    }
}
class WriterThread implements Runnable{

    protected BlockingQueue<String> blockingQueue = null;

    public WriterThread(BlockingQueue<String> blockingQueue){
        this.blockingQueue = blockingQueue;
    }

    @Override
    public void run() {
        PrintWriter writer = null;

        try {
            writer = new PrintWriter(new File("output.txt"));

            while(true){
                String buffer = blockingQueue.take();
                //Check whether end of file has been reached
                if(buffer.equals("EOF")){
                    break;
                }
                writer.println(buffer);
            }


        } catch (FileNotFoundException e) {

            e.printStackTrace();
        } catch(InterruptedException e){

        }finally{
            writer.close();
        }

    }

}
```