1. Text File I/O

In Java we use File class to load the file and Scanner class to scan through the content of that file. To get next line of the file we use nextLine() method.
In C++ we use ifstream to load and scan the file. To get next line we use getLine() function.

2. Support for Data Abstraction

Java provides abstraction through abstract classes and interfaces whereas C++ provides abstraction through classes, header files and access specifiers.

3. Class Libraries

Java and C++ provide a lot of inbuilt class libraries which developers can use to implement some functionality easily.
To import a library in Java, we use

```
            import java.library_name.class_name;
Example:    import java.util.Scanner;
```

To import a library in C++, we usr

```
            #include<library_name>
Example:    #include <cstdlib>
```

4. Random Number Generation

To generate random number in Java, we use Random class from util library. Code is

```
      import java.util.Random;
      Random random = new Random();
      int number = random.nextInt(10); // Get a random number between 0 to 9
```

To generate random number in C++, we use rand() function from cstdlib library. However, to get a new number every time we need to do one more line. That is,

```
      srand(time(0));
```

If we don't do this, we will get same number for every rand() call. Code is,

```
      #include <cstdlib>
      int number = rand() % 10 // Get a random number between 0 to 9
```

5. Exception Handling

To handle exceptions in a code in Java/C++ we wrap the code in a try {} block followed by catch(){} block which catches exception occurred in try{} block and runs code according to the error occurred.

```
      try{
            // Code
      }
      catch(Exception e){
            // Exception Handling Code
      }
      finally{
            // Code to run if we get error or not
```

```
}
```

6. String Manipulation

In both Java and C++, string is a class. We can’t directly manipulate the characters present in a string. If we change characters we get a new instance of string with modified version.

7. Command Line Arguments

In Java, we have a main() method inside the class which gets called when we run the file. That main() method takes a parameter String[] args which stores the command line arguments we send when we run the file.

```
public static void main(String[] args){
        for (String val : args)
                // Printing command line arguments
                System.out.println(val);
}
```

In C++, main() function can take 2 parameters argc, argv where argc stores number of arguments passed and argv has the actual arguments passed.

```
int main(int argc, char** argv) {
        cout << “Number of arguments:  " << argc << "\n";
        for (int i = 0; i < argc; ++i)
                cout << argv[i] << "\n";
        return 0;
}
```

8. Dynamic Storage Allocation

In Java, everything gets allocated to memory dynamically.

In C++, to allocate a memory dynamically we use new operator. To delete those variables we use delete operator to deallocate the memory we allocated.