1. Lambda Expressions
2. Functional interfaces
3. Introduced default methods in interfaces (we can declare concreate methods inside interfaces using default method) –From Java 1.8 versions
4. We can declare static methods in interfaces–From Java 1.8 versions
5. There are some predefined functional interfaces: Predicate (we can use in Lambda expressions)
6. There are some predefined functional interfaces: Function (we can use in Lambda expressions)
7. There are some predefined functional interfaces: Consumer (we can use in Lambda expressions)
8. Method Reference and Constructor reference by using :: . Scope resolution operator (::) operator (this is already there in C++, but the purpose is different)
9. Stream API: To perform bulk operations on collections, we can use streams concepts.
10. Date and time API to handle date and time (introduce by joda.org). this is also known as Joda API

Purpose of Java 8 (main intention of java 8)

1. To simplify programming

Simple and easy to write (more concise version). To provide easiness to programmer. It simplifies programmer’s life. Try learning new features than thinking oh! Again, new features

More concise clean code

1. To utilize functional programming benefits in Java Java is Object oriented programming. It’s not functional oriented language. There are many benefits in functional oriented programming language. We are missing them in object-oriented programming, that’s why Java 8 introduced functional programming. To enable functional programming, Java 8 introduced Lambdas. Java is still Object-oriented programming (even we introduced functional programming).
2. To enable parallel programming in Java (it uses multi cores).