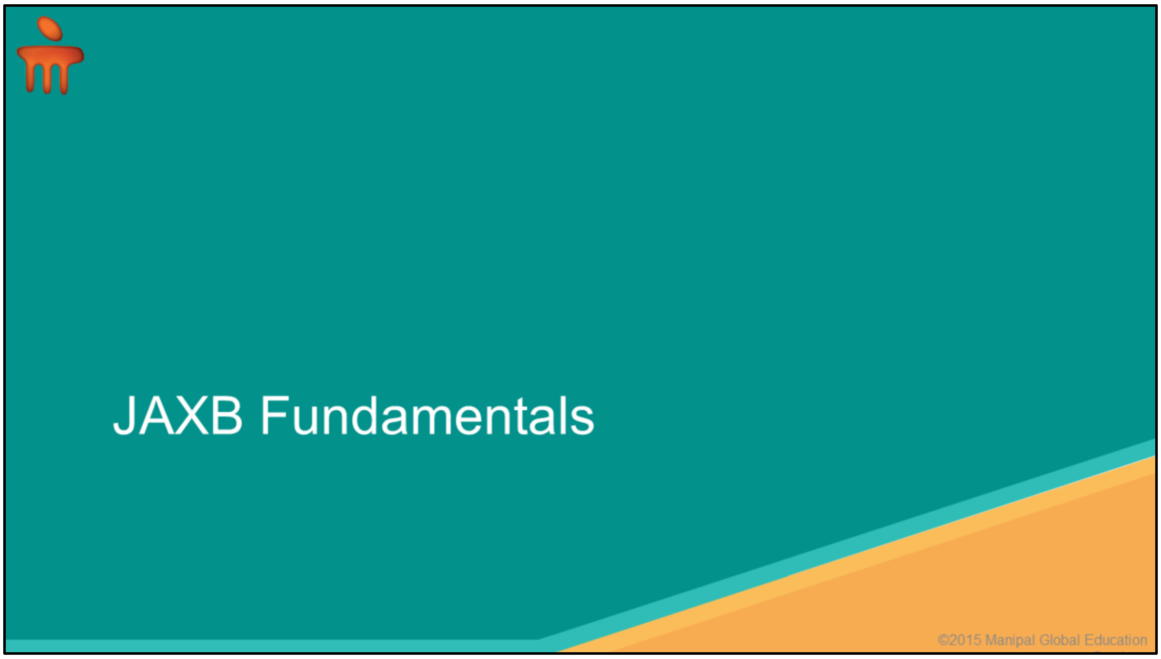
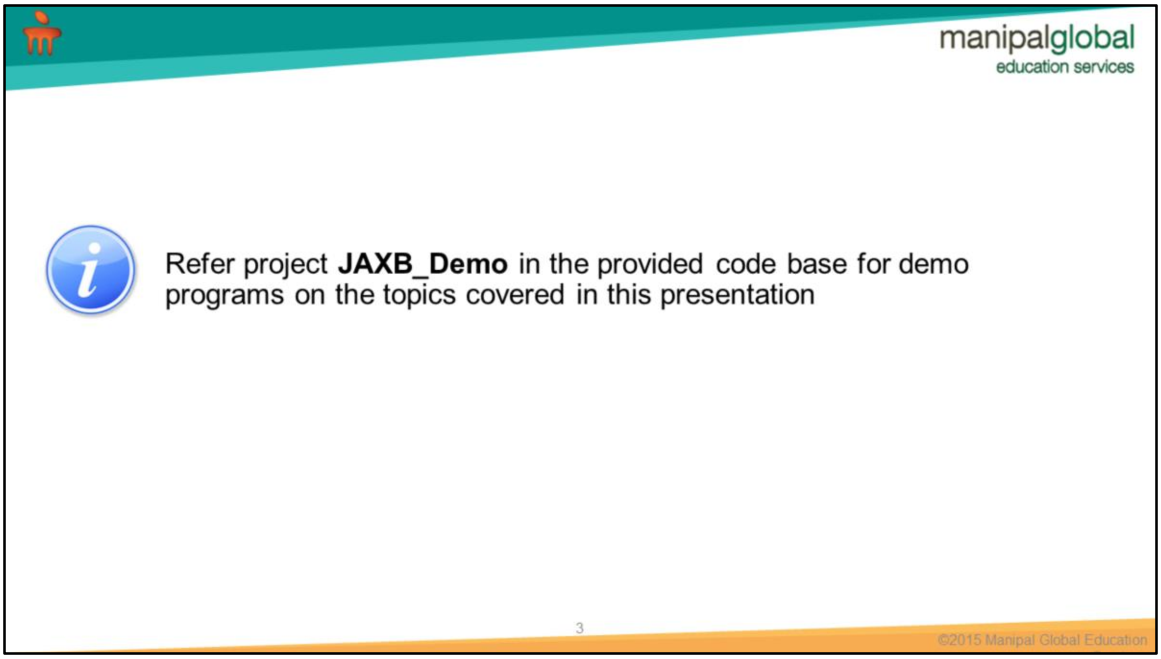


JAXB Fundamentals

LEARNING OBJECTIVES

At the end of this lesson, you will be able to:

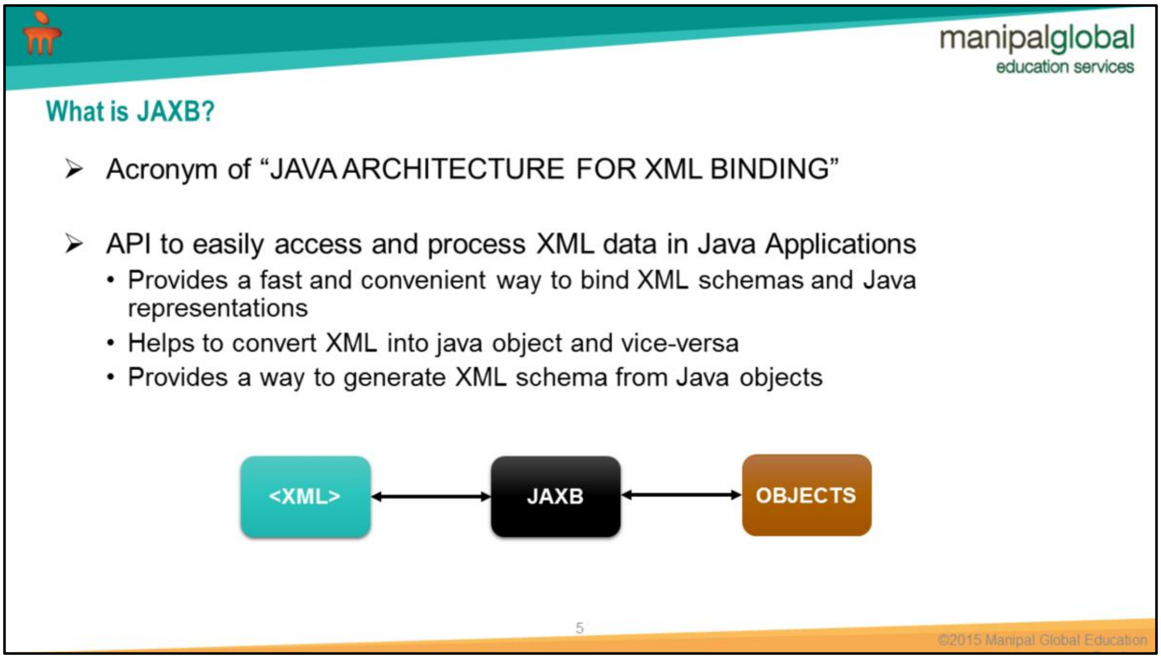
- ○ Describe JAXB functionalities
- ○ Define Marshalling and UnMarshalling
- ○ List the Steps and demonstrate Marshalling
- ○ List the Steps and demonstrate UnMarshalling



Refer project **JAXB_Demo** in the provided code base for demo programs on the topics covered in this presentation

CONCEPT

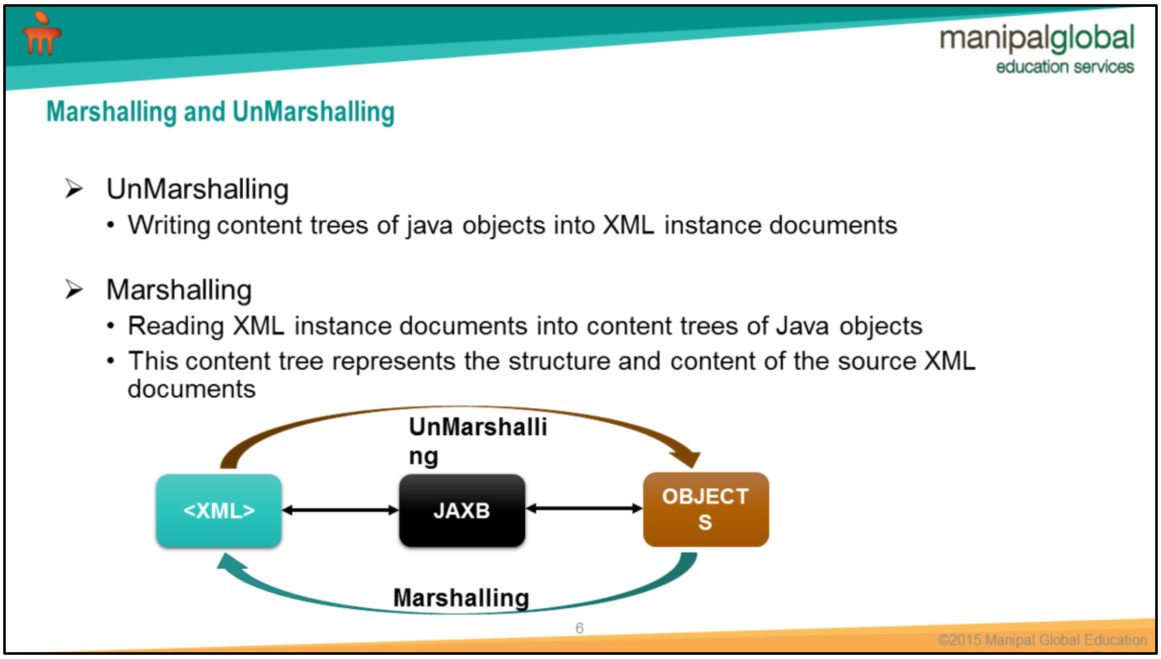
JAXB Fundamentals



What is JAXB?

- Acronym of “JAVA ARCHITECTURE FOR XML BINDING”
- API to easily access and process XML data in Java Applications
  - Provides a fast and convenient way to bind XML schemas and Java representations
  - Helps to convert XML into java object and vice-versa
  - Provides a way to generate XML schema from Java objects

<XML>
JAXB
OBJECTS



manipalglobal
education services

Marshalling and UnMarshalling

- UnMarshalling
  - Writing content trees of java objects into XML instance documents
- Marshalling
  - Reading XML instance documents into content trees of Java objects
  - This content tree represents the structure and content of the source XML documents

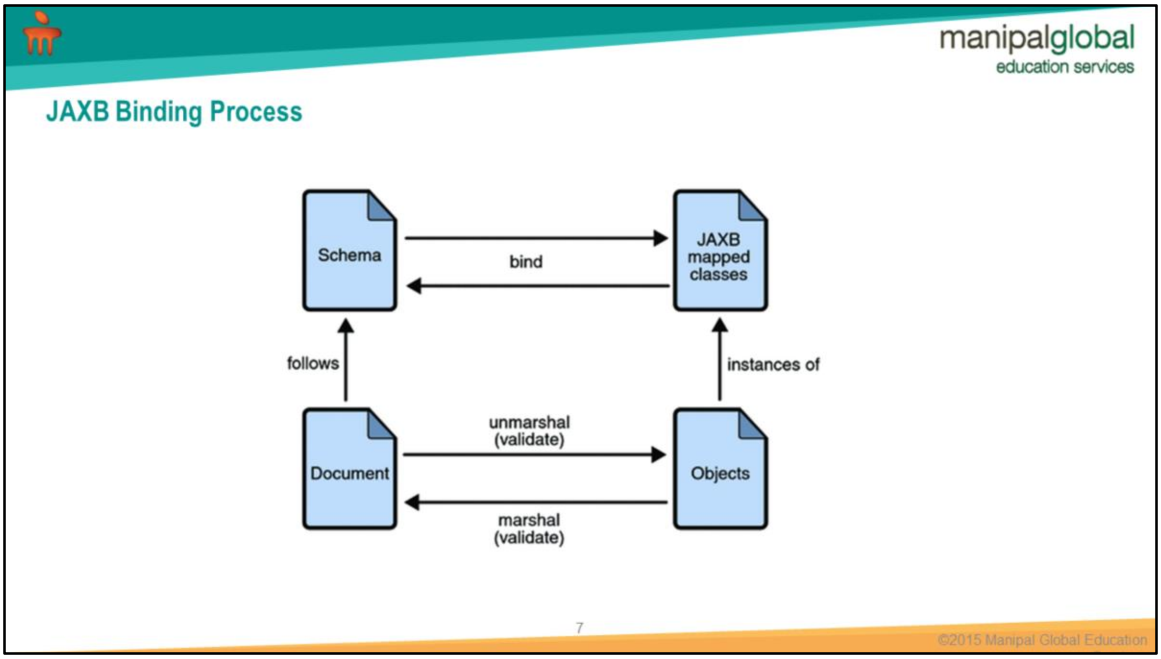
UnMarshalli
ng

<XML>

JAXB

OBJECT
S

Marshalling



manipalglobal
education services

JAXB Binding Process

7

The general steps in the JAXB data binding process are:

Generate classes:
An XML schema is used as input to the JAXB binding compiler to generate JAXB classes based on that schema.
Compile classes:
All of the generated classes, source files, and application code must be compiled.
Unmarshal:
XML documents written according to the constraints in the source schema are unmarshalled by the JAXB binding framework.
Note that JAXB also supports unmarshalling XML data from sources other than files and documents, such as DOM nodes, string buffers, SAX sources, and so forth.
Generate content tree:
The unmarshalling process generates a content tree of data objects instantiated from the generated JAXB classes;
this content tree represents the structure and content of the source XML documents.
Validate (optional):
The unmarshalling process involves validation of the source XML documents before generating the content tree.

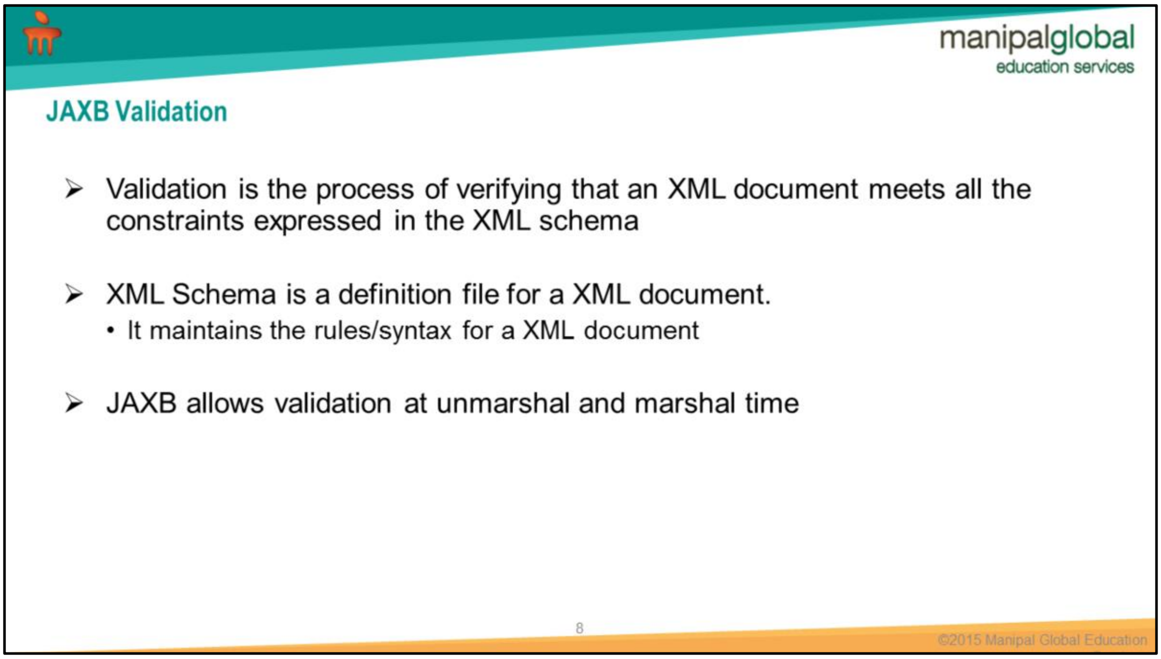
Note that if you modify the content tree in Step 6, you can also use the JAXB Validate operation to validate the changes before marshalling the content back to an XML document.

Process content:

The client application can modify the XML data represented by the Java content tree by using interfaces generated by the binding compiler.

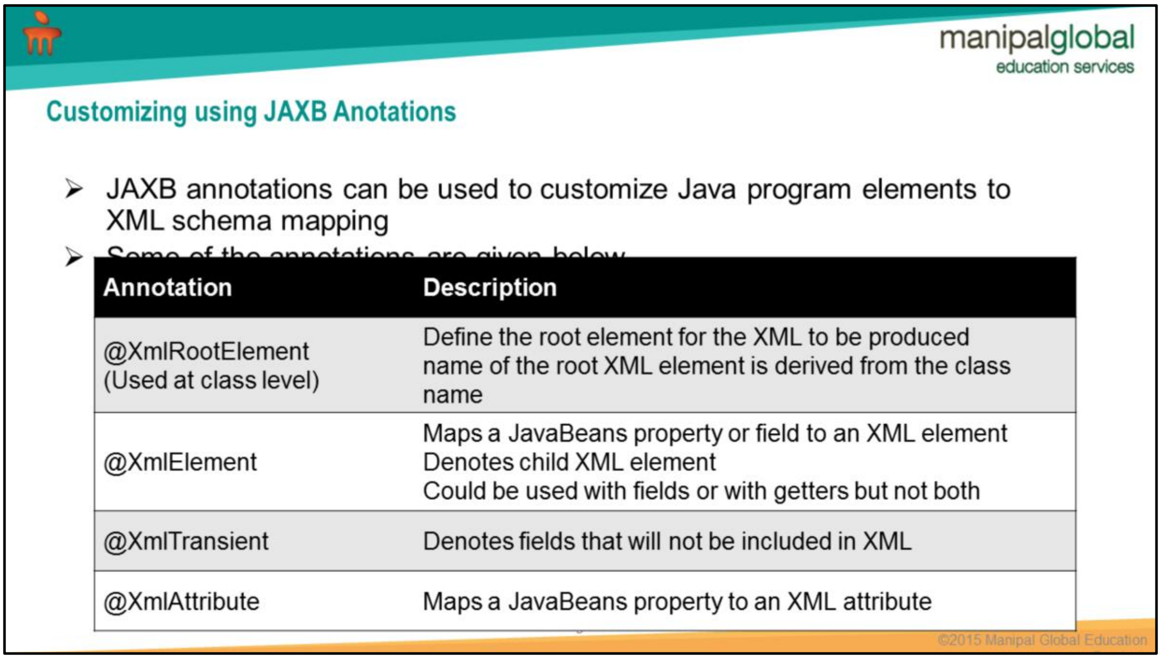
Marshal:

The processed content tree is marshalled out to one or more XML output documents. The content may be validated before marshalling.



JAXB Validation

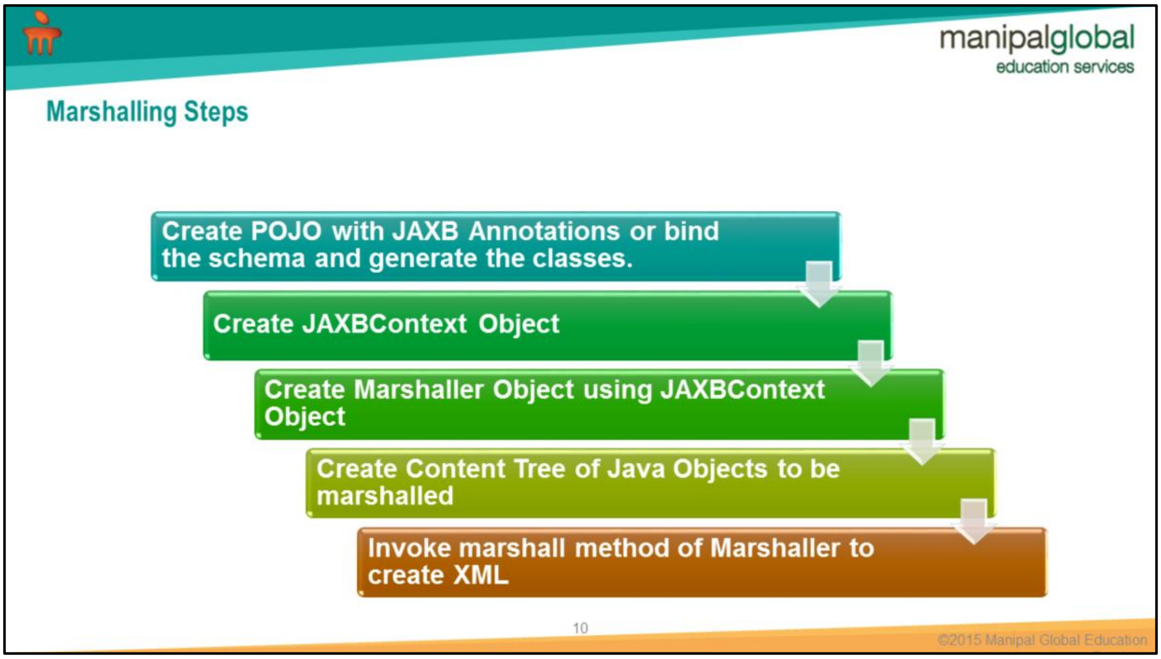
- Validation is the process of verifying that an XML document meets all the constraints expressed in the XML schema
- XML Schema is a definition file for a XML document.
  - It maintains the rules/syntax for a XML document
- JAXB allows validation at unmarshal and marshal time



Customizing using JAXB Anotations

- JAXB annotations can be used to customize Java program elements to XML schema mapping
- Some of the annotations are given below

Annotation	Description
@XmlRootElement (Used at class level)	Define the root element for the XML to be produced name of the root XML element is derived from the class name
@XmlElement	Maps a JavaBeans property or field to an XML element Denotes child XML element Could be used with fields or with getters but not both
@XmlTransient	Denotes fields that will not be included in XML
@XmlAttribute	Maps a JavaBeans property to an XML attribute



manipalglobal
education services

Marshalling Steps

Create POJO with JAXB Annotations or bind the schema and generate the classes.

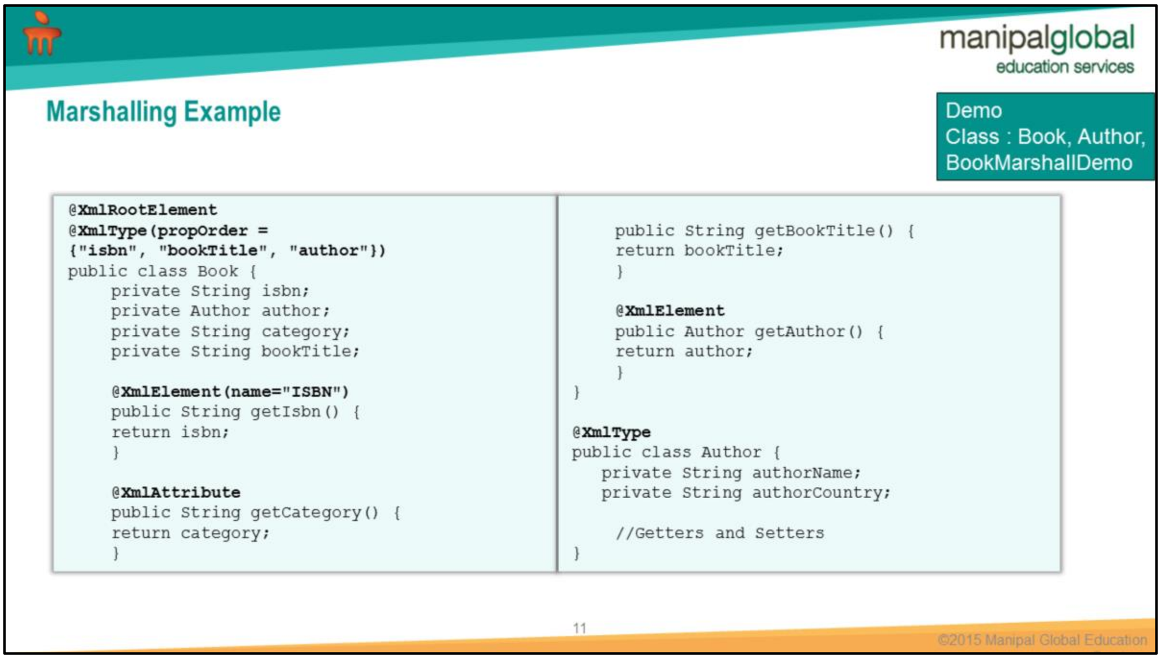
Create JAXBContext Object

Create Marshaller Object using JAXBContext Object

Create Content Tree of Java Objects to be marshalled

Invoke marshall method of Marshaller to create XML

10



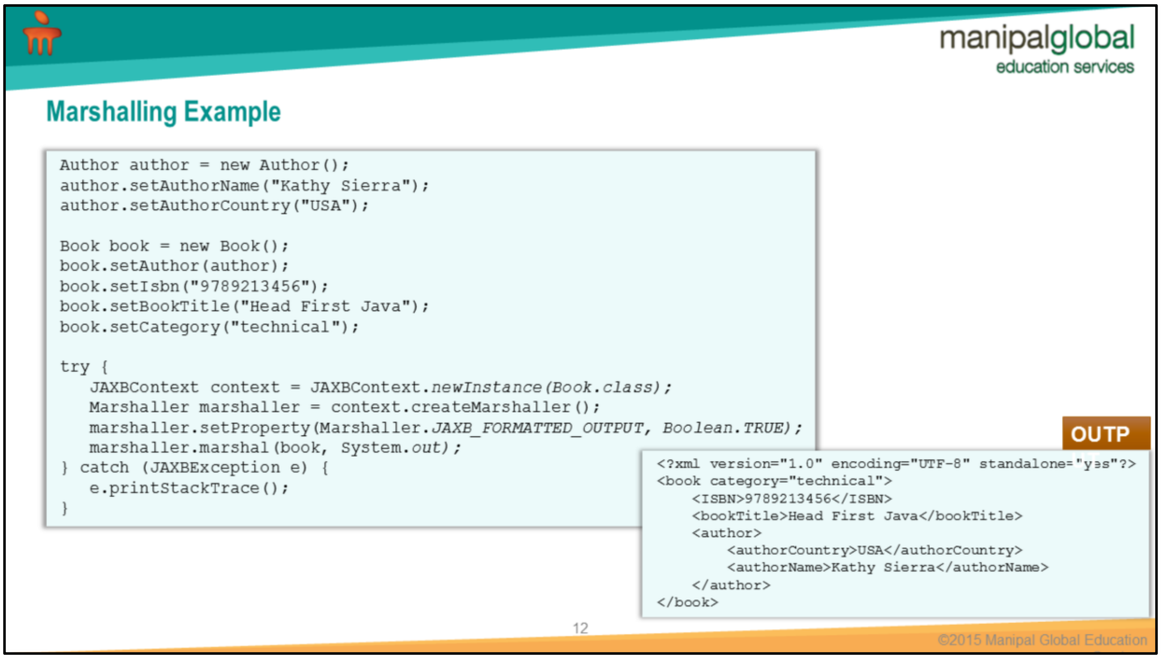
manipalglobal
education services

Marshalling Example

Demo
Class : Book, Author, BookMarshallDemo

```
@XmlRootElement
@XmlType(propOrder =
{"isbn", "bookTitle", "author"})
public class Book {
    private String isbn;
    private Author author;
    private String category;
    private String bookTitle;

    @XmlElement(name="ISBN")
    public String getIsbn() {
    return isbn;
    }

    @XmlAttribute
    public String getCategory() {
    return category;
    }
```

```
    public String getBookTitle() {
    return bookTitle;
    }

    @XmlElement
    public Author getAuthor() {
    return author;
    }
}

@XmlType
public class Author {
   private String authorName;
   private String authorCountry;

    //Getters and Setters
}
```



manipalglobal
education services

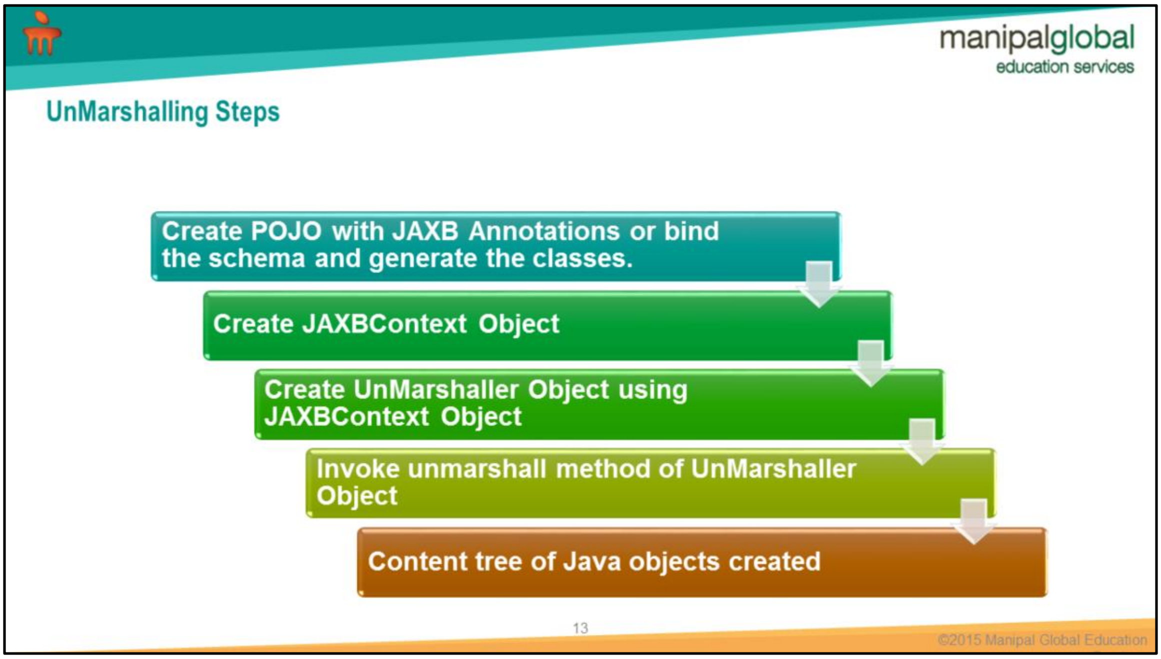
Marshalling Example

```
Author author = new Author();
author.setAuthorName("Kathy Sierra");
author.setAuthorCountry("USA");

Book book = new Book();
book.setAuthor(author);
book.setIsbn("9789213456");
book.setBookTitle("Head First Java");
book.setCategory("technical");

try {
   JAXBContext context = JAXBContext.newInstance(Book.class);
   Marshaller marshaller = context.createMarshaller();
   marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, Boolean.TRUE);
   marshaller.marshal(book, System.out);
} catch (JAXBException e) {
   e.printStackTrace();
}
```

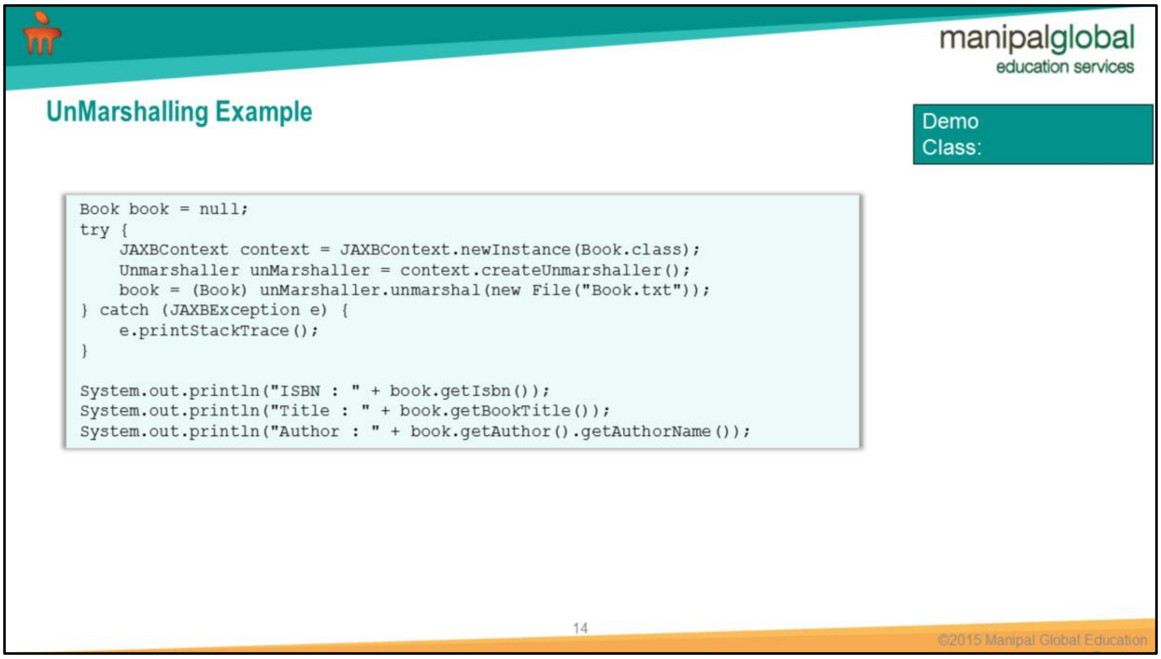
OUTP

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<book category="technical">
    <ISBN>9789213456</ISBN>
    <bookTitle>Head First Java</bookTitle>
    <author>
        <authorCountry>USA</authorCountry>
        <authorName>Kathy Sierra</authorName>
    </author>
</book>
```

UnMarshalling Steps



1. Create POJO with JAXB Annotations or bind the schema and generate the classes.
2. Create JAXBContext Object
3. Create UnMarshaller Object using JAXBContext Object
4. Invoke unmarshall method of UnMarshaller Object
5. Content tree of Java objects created



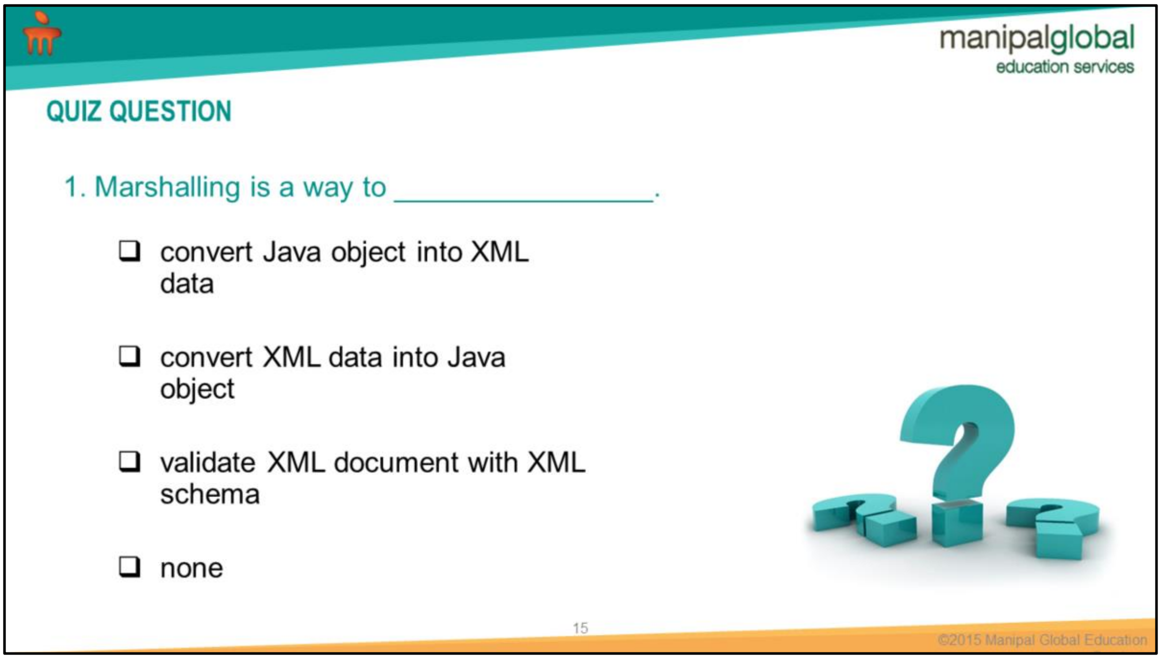
manipalglobal
education services

UnMarshalling Example

Demo
Class:

```
Book book = null;
try {
    JAXBContext context = JAXBContext.newInstance(Book.class);
    Unmarshaller unMarshaller = context.createUnmarshaller();
    book = (Book) unMarshaller.unmarshal(new File("Book.txt"));
} catch (JAXBException e) {
    e.printStackTrace();
}

System.out.println("ISBN : " + book.getIsbn());
System.out.println("Title : " + book.getBookTitle());
System.out.println("Author : " + book.getAuthor().getAuthorName());
```

14

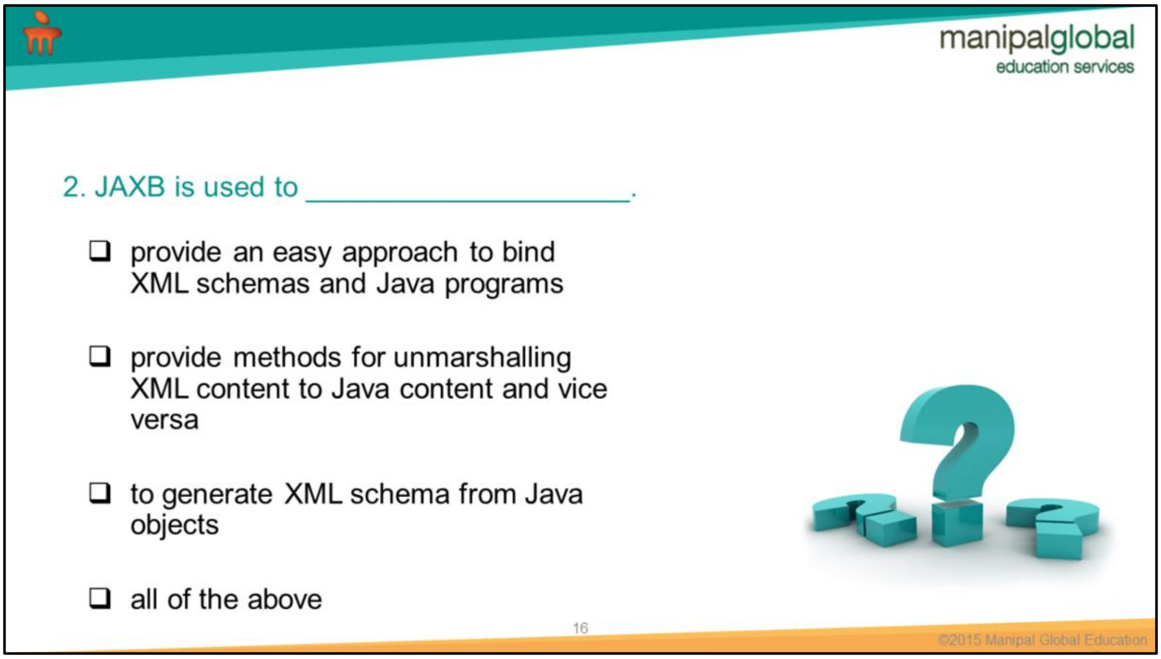


QUIZ QUESTION

1. Marshalling is a way to ________________.

- □ convert Java object into XML data
- □ convert XML data into Java object
- □ validate XML document with XML schema
- □ none

Answer: convert Java object into XML data



manipalglobal
education services

2. JAXB is used to ____________________.

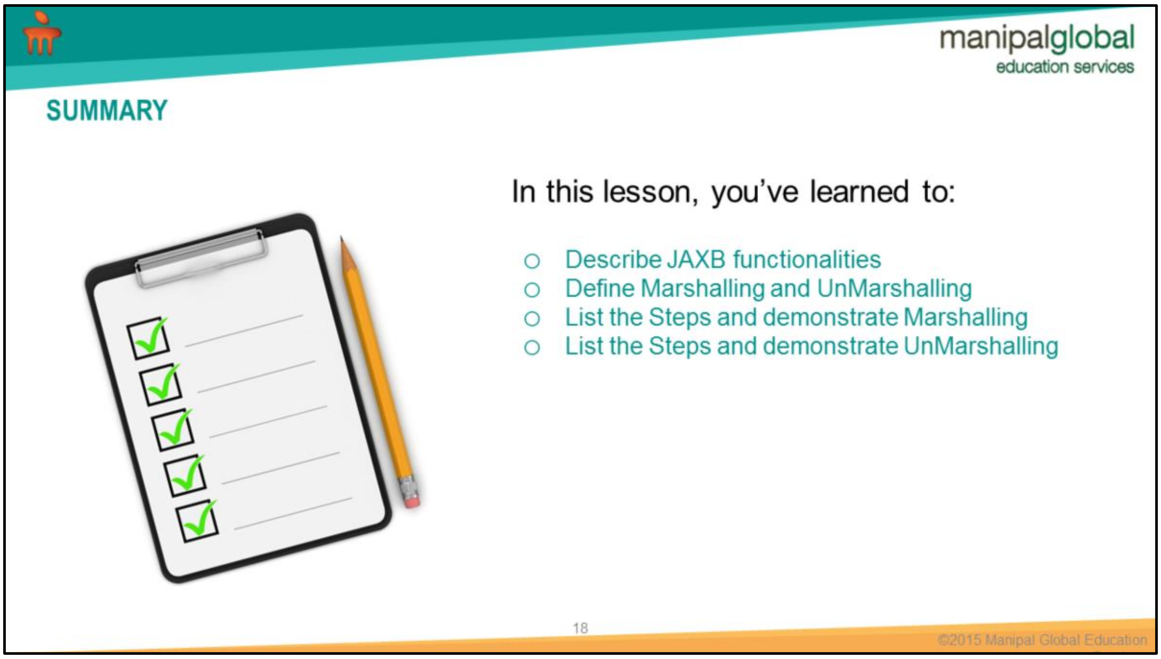
- ❑ provide an easy approach to bind XML schemas and Java programs
- ❑ provide methods for unmarshalling XML content to Java content and vice versa
- ❑ to generate XML schema from Java objects
- ❑ all of the above

16

Answer: all of the above

SUMMARY

Introduction to JAXB



manipalglobal
education services

SUMMARY

In this lesson, you've learned to:

- ○ Describe JAXB functionalities
- ○ Define Marshalling and UnMarshalling
- ○ List the Steps and demonstrate Marshalling
- ○ List the Steps and demonstrate UnMarshalling