Introduction to Java

Introduction

Who am I?

Expectations

What are we getting out of this workshop?

Java in our day-to-day lives

- www.amazon.com, www.ebay.com, www.linkedin.com
- 100% of Blue Ray devices ship with Java
- Phones, Cars, Home Automation

Fun Facts

Java was first named “Oak”, after a tree outside James Golsling’s home

9 million java developers

3 Billion devices run Java

Programming Basics

1. Understanding the problem at a high level.

2. Write flowcharts – First step from idea towards programming.

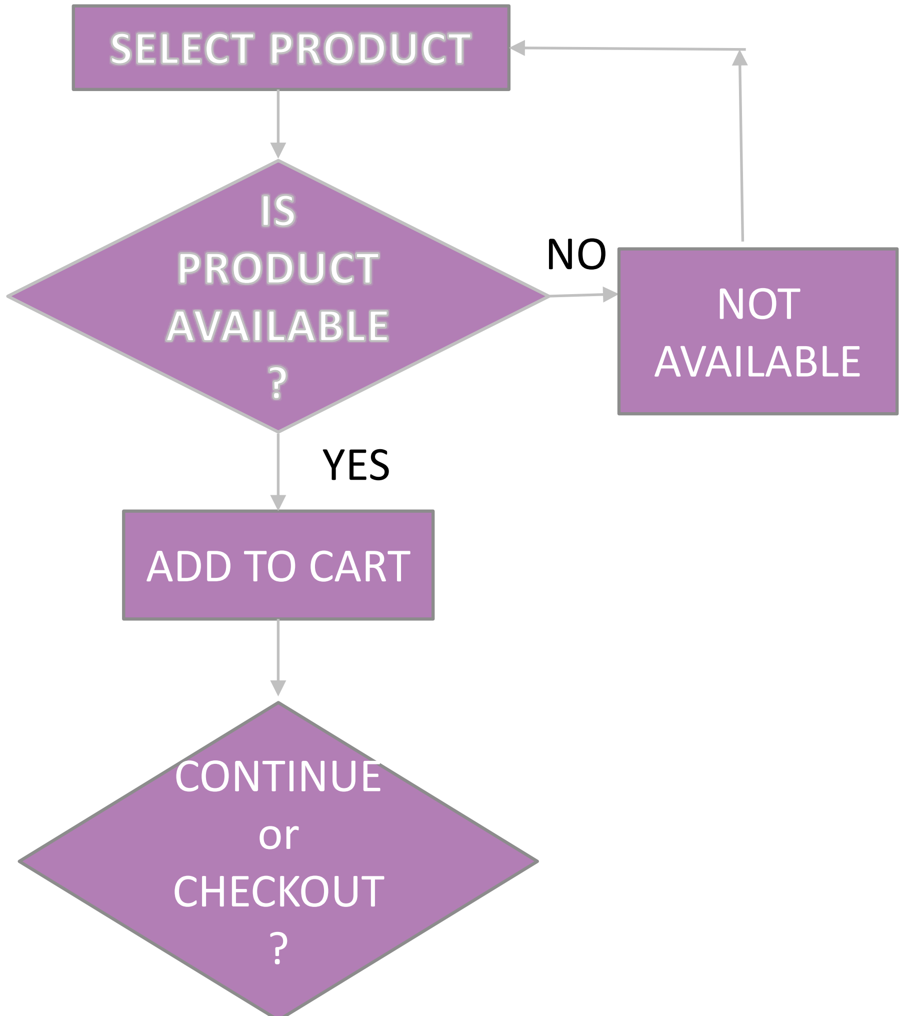
3. Write pseudo code – Textual representation that’s closer to code, but not tied to a programming language, no need to get caught up with syntaxes.

4. Write code in the programming language.

5. Test cases and see if the solution works for all use cases

Flowchart / Pseudocode

FLOWCHART

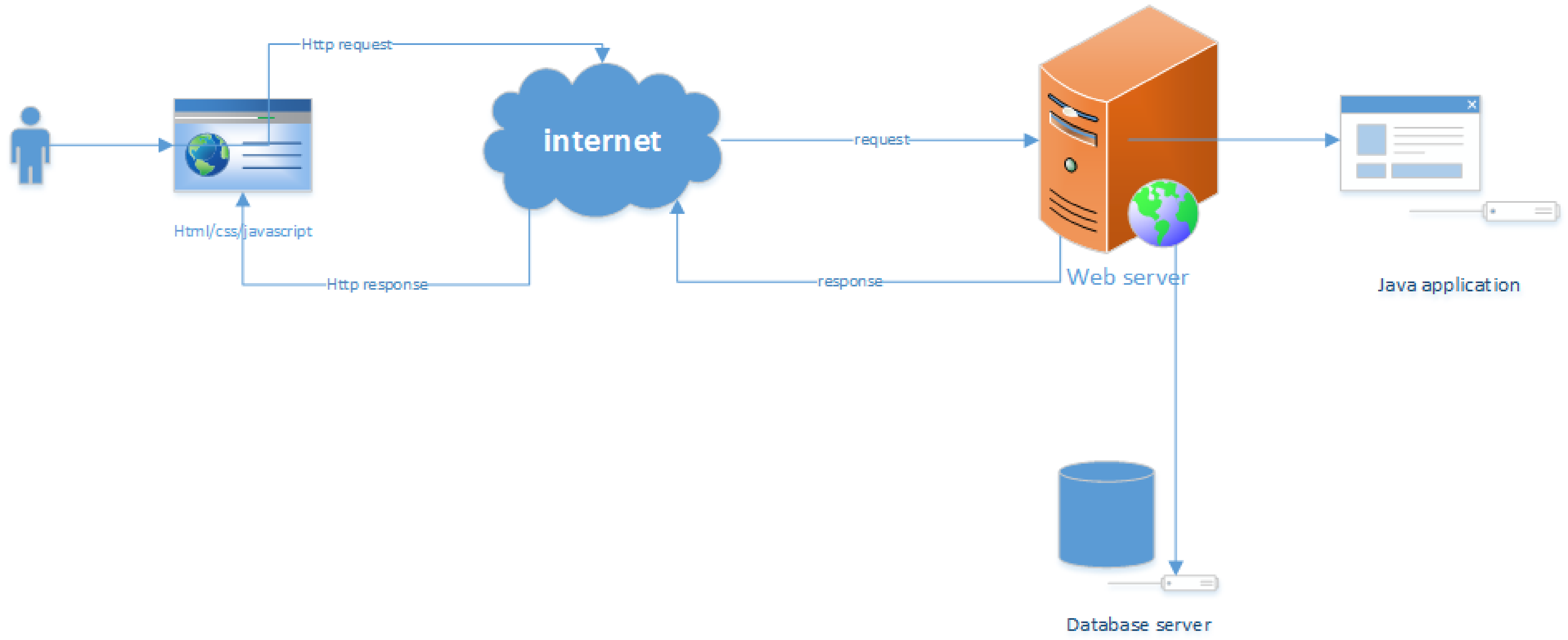


PSEUDOCODE

```
Select Product

If product is available
    Add the product to user's cart
    Continue Shopping or Checkout

If product is not available
   Prompt the user to select another product

Repeat the same process for the new product
selected by the user
```

Web Architecture

1. The browser(client) makes a request to the server over the internet using the HTTP protocol.
2. The web server then can construct the data by accessing the application/database servers and returns a response.
3. The response from the web server is HTML that the browser renders.



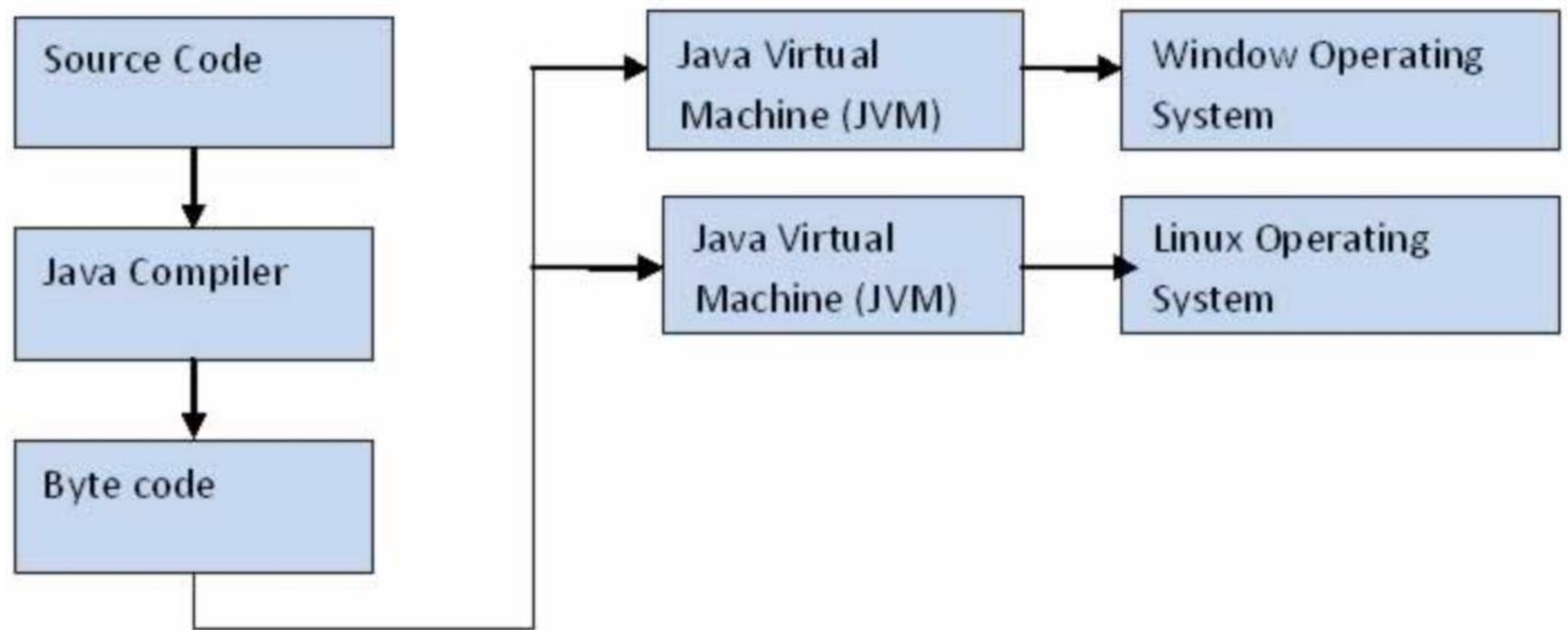
Java Program

Computer program is set if instructions telling the computer what to do.

Java Compiler compiles (translates) the code you have written to Bytecode.

The compiled program can then be run on any computer that has an interpreter for the Java virtual machine. Other languages have to be re-compiled for each platform on which they are going to run.

Portability – Write once, run anywhere



A Simple Java program

```
public static void main(String args[])
{
 System.out.println("Chocolate, royalties, sleep");
}
```

main: The main starting point for execution in every Java program.

String: A bunch of text; a row of characters, one after another.

System.out.println: This method is defined in the Java API. Whenever you call the System.out.println method, the computer displays text on its screen

Java Methods

Method is a list of things to do.

1. **Method Declaration** - Defined the name of the method, what type its returning and the list of tasks to be performed by the method.

```
 int addNumbers(int num1, int num2) {
         //Code to add 2 numbers
... }
```

2. **Method Invocation** – Java program needs to have an instruction to call the addToCart method into action. That’s called method invocation.

```
{……
addNumbers(3,4);
….}
```

Java Methods Example

```
class Main {
      public static void main(String[] args) {
            System.out.println("hello world");
            doSomething();
      }
      static void doSomething(){
            // this is comment
            System.out.println("hi i'm in the doSomething method");
      }
}
```

indentation

comment

Semi colon

Types

Type of the content you want to store

Primitive Types: byte, short, int, long, float, double

Each of these types has its own range of possible values. Eg. Int –2147483648 to 2147483647

int val = 12;

float val1 = 12.4F;

double val2 = 12.4;

long val3 = 123456789L;

Char and String:

Char c = ‘a‘;

String s = “abc“;

Boolean:

true or false

Variables, Values

- A **variable** is a placeholder for storing content. You can storea number like 5 into a variable.
- The thing stored in a variable is called a **value**. For Example if Linda's age is 14, in Java it would be represented like int age = 14;
- Assigning values to variables - read form right to left. **int amount = 5** - Assign 5 to the amount variable
- **Exercise:**

```
class MyTestClass
{
public static void main(String args[]) {
        int amount = 5;
        System.out.println("The amount is" + amount);
} }
```

Operators

Operators **:**

+, *, - , %. Increment (++), decrement (--)

Exercise:

```
class MyTestClass
{
public static void main(String args[]) {
        int amount = 5;
        System.out.print(“The amount is: ” + amount);
        amount = amount++;
        System.out.println(“The new amount is: ” + amount);
        amount = amount * 4;
        System.out.println(“The new amount is now:  ” + amount);
} }
```

Flow Control

If Else statements

```
if (Condition)
      { SomeStatements }
 else
     { OtherStatements }    (Exercise – If amount is even, print even else print odd)
```

While - (Repeating instructions over and over as long as a Condition is met)

```
while (Condition)
     { Statements }
```

Flow Control Contd…

For loop

```
for(number of times to repeat)
{
        Perform operation
}
```

(**Exercise – Print "hello" 5 times**)

Switch-Case

Switch-Case: Multiple execution paths based on the expression evaluation

```
Switch(expression)
{
    case "value1":
        Statements;
        break;
    case "value2" :
        Statements;
        break;
    case "default":
        Statements;
        break;
}
```

Object Oriented Programming Concepts

Class is an abstraction that defines attributes/properties and behaviors of real world objects like cars

Object is a particular instance of a Class

```
public class Car {
   String color;
   public void startCar() {
   //Instructions to start car
}
```

CLASS : CAR		
METHOD *startCar()*	**what it does:** starts the car	**PROPERTIES:** ***Make***: BMW ***Color***: blue ***Fuel***: diesel

Exercise: Can you come up with some real world objects you see in your day-to-day life? Lets model some Classes.

Example of Class and Object

Car Class:

```
public class Car {
String color;
public void startCar() {
    //Instructions to start car
}
public void setColor(String color) {
    this.color = color;
}
}
```

Creating Car Object:

```
Car car = new Car();
car.setColor("red");
car.startCar();
```

Fields, Methods, Constructors

Fields store data for each object to use.

Constructors allow each object to be set up properly when it is first created.

Methods implement the behavior of the object.

```
public class ClassName
{
Fields
Constructors
Methods
}
```

Exercise: (Create a Car Class, that has attributes for color, make and model. And then create 3 Car Objects with desired values for Color, make and model for each of them.)

```
public class Car{
private String color;
private String make;
public Car(String color, String make) { //Constructor
        this.color = color;
        this.make = make;
}
public String getColor() { //method declaration
        return color;
}
public void printInfo() { // method declaration
        System.out.println("My color of my car is " + getColor()); // method invocation
}
public class XYZ {
public static void main() {
        Car car = new Car("red", "Toyota"); //Creating Car object, invoking constructor
}
```

Online Resources

Code Academy

https://www.codecademy.com/learn/learn-java

Lynda

https://www.lynda.com/in/Java

Udemy

https://www.udemy.com/javaprogrammingcourse/

Beginner’s Tutorial

http://javabeginnerstutorial.com/core-java/