1. package Day7;

public class BooleanWrapperDemo {

public static void main(String[] args) {

Boolean b1 = Boolean.*valueOf*(true); // from boolean primitive

Boolean b2 = Boolean.*valueOf*("true"); // from String

System.***out***.println("b1: " + b1);

System.***out***.println("b2: " + b2);

// Boolean parsing

if (b1) {

System.***out***.println("b1 is true");

}

}

}

2. package Day7;

class Box {

int length;

Box(int length) {

this.length = length;

}

}

public class BoxDemo {

public static void modifyLength(Box b) {

b.length = 50; // modifying the object

}

public static void main(String[] args) {

Box box = new Box(10);

System.***out***.println("Before modification: " + box.length);

*modifyLength*(box);

System.***out***.println("After modification: " + box.length);

}

}

3. package Day7;

public class CallByReference {

public static void increment(int[] num) {

num[0] = num[0] + 1;

}

public static void main(String[] args) {

int[] x = {10}; // array wrapper

System.***out***.println("Before increment: " + x[0]);

*increment*(x);

System.***out***.println("After increment: " + x[0]);

}

}

4. package Day7;

class Box2 {

int length;

Box2(int length) { this.length = length; }

}

public class CallByValueDemo {

public static void changeObject(Box2 b) {

b = new Box2(100); // assign new object

}

public static void main(String[] args) {

Box2 box = new Box2(10);

System.***out***.println("Before method: " + box.length);

*changeObject*(box);

System.***out***.println("After method: " + box.length);

}

}

5. package Day7;

public class Char\_digit\_wrapper {

public static void main(String[] args) {

char ch = '5';

if (Character.*isDigit*(ch)) {

System.***out***.println(ch + " is a digit.");

} else {

System.***out***.println(ch + " is not a digit.");

}

}

}

6. package Day7;

public class NullWrapperDemo {

public static void main(String[] args) {

Integer intObj = null;

Double doubleObj = null;

Boolean boolObj = null;

System.***out***.println("Integer object: " + intObj);

System.***out***.println("Double object: " + doubleObj);

System.***out***.println("Boolean object: " + boolObj);

}

}

7. package Day7;

public class PassByValueDemo1 {

public static void main(String[] args) {

int num = 10;

System.***out***.println("Before method call: " + num);

*changeValue*(num); // pass primitive

System.***out***.println("After method call: " + num);

}

static void changeValue(int n) {

n = 50; // change local copy

System.***out***.println("Inside method: " + n);

}}

8. package Day7;

public class PassByValueDemo2 {

public static void main(String[] args) {

int a = 5, b = 10;

System.***out***.println("Before swap: a = " + a + ", b = " + b);

*swap*(a, b); // attempt to swap

System.***out***.println("After swap: a = " + a + ", b = " + b);

}

static void swap(int x, int y) {

int temp = x;

x = y;

y = temp;

System.***out***.println("Inside swap method: x = " + x + ", y = " + y);

}

}

9. package Day7;

public class PassByValueDemo3 {

public static void main(String[] args) {

int x = 20;

double y = 15.5;

boolean flag = true;

System.***out***.println("Before method call: x=" + x + ", y=" + y + ", flag=" + flag);

*modifyPrimitives*(x, y, flag);

System.***out***.println("After method call: x=" + x + ", y=" + y + ", flag=" + flag);

}

static void modifyPrimitives(int a, double b, boolean c) {

a = 100;

b = 200.5;

c = false;

System.***out***.println("Inside method: a=" + a + ", b=" + b + ", c=" + c);

}}

10. package Day7;

public class StringCompareDemo {

public static void main(String[] args) {

String str1 = "Hello";

String str2 = "hello";

// Using equals()

if (str1.equals(str2)) {

System.***out***.println("Strings are equal (case-sensitive).");

} else {

System.***out***.println("Strings are not equal (case-sensitive).");

}

// Using equalsIgnoreCase()

if (str1.equalsIgnoreCase(str2)) {

System.***out***.println("Strings are equal (ignore case).");

}

}

}

11. package Day7;

class Student {

String name;

int marks;

Student(String name, int marks) {

this.name = name;

this.marks = marks;

}

}

public class StudentDemo {

public static void updateMarks(Student s, int newMarks) {

s.marks = newMarks;

}

public static void main(String[] args) {

Student student = new Student("Alice", 85);

System.***out***.println("Before update: " + student.name + " - " + student.marks);

*updateMarks*(student, 95);

System.***out***.println("After update: " + student.name + " - " + student.marks);

}

}

12. package Day7;

class MyThread1 extends Thread {

public void run() {

for (int i = 1; i <= 5; i++) {

System.***out***.println("Thread: " + i);

}

}

}

public class ThreadDemo1 {

public static void main(String[] args) {

MyThread1 t = new MyThread1();

t.start();

}

}

13. package Day7;

class CounterThread extends Thread {

static int *counter* = 0;

public void run() {

for (int i=0;i<1000;i++) *counter*++;

}

}

public class ThreadDemo10 {

public static void main(String[] args) throws InterruptedException {

CounterThread t1 = new CounterThread();

CounterThread t2 = new CounterThread();

t1.start(); t2.start();

t1.join(); t2.join();

System.***out***.println("Counter: " + CounterThread.*counter*);

}

}

14. package Day7;

class MyRunnable implements Runnable {

public void run() {

System.***out***.println("Current Thread: " + Thread.*currentThread*().getName());

}

}

public class ThreadDemo2 {

public static void main(String[] args) {

Thread t = new Thread(new MyRunnable());

t.start();

}

}

15. package Day7;

class MessageThread extends Thread {

String message;

MessageThread(String message) { this.message = message; }

public void run() {

for (int i = 0; i < 5; i++) System.***out***.println(message);

}

}

public class ThreadDemo3 {

public static void main(String[] args) {

new MessageThread("Hello").start();

new MessageThread("World").start();

}

}

16. package Day7;

class SleepDemo extends Thread {

public void run() {

for (int i = 1; i <= 3; i++) {

System.***out***.println(i);

try { Thread.*sleep*(1000); } catch (InterruptedException e) {}

}

}

}

public class ThreadDemo4 {

public static void main(String[] args) {

new SleepDemo().start();

}

}

17. package Day7;

class YieldDemo extends Thread {

public void run() {

for (int i = 1; i <= 5; i++) {

System.***out***.println(getName() + ": " + i);

Thread.*yield*();

}

}

}

public class ThreadDemo5 {

public static void main(String[] args) {

YieldDemo t1 = new YieldDemo();

YieldDemo t2 = new YieldDemo();

t1.start(); t2.start();

}

}

18. package Day7;

class EvenThread extends Thread {

public void run() {

for (int i = 2; i <= 10; i+=2) System.***out***.println("Even: " + i);

}

}

class OddThread extends Thread {

public void run() {

for (int i = 1; i <= 9; i+=2) System.***out***.println("Odd: " + i);

}

}

public class ThreadDemo6 {

public static void main(String[] args) {

new EvenThread().start();

new OddThread().start();

}

}

19. package Day7;

class PriorityThread extends Thread {

PriorityThread(String name) { super(name); }

public void run() { System.***out***.println(getName() + " running with priority " + getPriority()); }

}

public class ThreadDemo7 {

public static void main(String[] args) {

PriorityThread t1 = new PriorityThread("T1"); t1.setPriority(Thread.***MIN\_PRIORITY***);

PriorityThread t2 = new PriorityThread("T2"); t2.setPriority(Thread.***NORM\_PRIORITY***);

PriorityThread t3 = new PriorityThread("T3"); t3.setPriority(Thread.***MAX\_PRIORITY***);

t1.start(); t2.start(); t3.start();

}

}

20. package Day7;

class JoinThread extends Thread {

public void run() {

for (int i=1;i<=3;i++) System.***out***.println(getName() + ": " + i);

}

}

public class ThreadDemo8 {

public static void main(String[] args) throws InterruptedException {

JoinThread t1 = new JoinThread(); t1.setName("T1");

JoinThread t2 = new JoinThread(); t2.setName("T2");

t1.start();

t1.join(); // wait for t1 to finish

t2.start();

}

}

21. package Day7;

class FlagThread extends Thread {

volatile boolean running = true;

public void run() {

while(running) System.***out***.println("Running...");

System.***out***.println("Thread stopped");

}

public void stopThread() { running = false; }

}

public class ThreadDemo9 {

public static void main(String[] args) throws InterruptedException {

FlagThread t = new FlagThread();

t.start();

Thread.*sleep*(1000);

t.stopThread();

}}

22. package Day7;

public class ValueOfDemo {

public static void main(String[] args) {

int num = 123;

double d = 45.67;

// Convert primitives to wrapper objects using valueOf

Integer intObj = Integer.*valueOf*(num);

Double doubleObj = Double.*valueOf*(d);

System.***out***.println("Integer object: " + intObj);

System.***out***.println("Double object: " + doubleObj);

}

}