



Uttara InfoSolutions

[www.uttarainfo.com](http://www.uttarainfo.com)

## **Uttara Practicals 2**

### **Instructions:**

**If you have not done the first lab or completed all its questions, I would strongly recommend you to first attempt the Lab1 and then only continue with this lab.**

**I would strongly recommend also that you spend extra time in the lab tomorrow and day after if required to complete both the Lab questions depending on your speed of learning.**

This lab focuses on helping you to understand working of operators further and how to apply the control statements you have learnt to solve simple programming problems.

Understand given Test.java and PassParamTest1.java (there are 3 class definitions with main() -> test each one of them) working given. Do you understand how the control is flowing? Ask doubts if you have any but first understand the control flow, passing of parameters, printing of values in main() and then start this lab. Also go through Hippo.java and TestHippo.java. Compile them and run and verify your understanding first.

If you are new to programming, I would recommend you do all the below mentioned problems, slowly but surely. Take assistance of Lab Instructors please.

**Very simple programs to help you to understand working of operators and precedence:**

- 1) WAP to assign an Hex value to an int variable and see the result. Similarly check it for Octal?
- 2) WAP to add two int values and see the result by printing it using SOP? Then check for doubles.
- 3) WAP add char variable with 'a' assigned to it with value 100 and see the result? Print it both as char as well as int value.
- 4) WAP to add byte and short value then assign the output value to short and see the result? Does it compile directly? What should you do to compile successfully?
- 5) WAP to assign a Maximum long(data type) value to float and see the result? (use long l = -1L >>> 1; to get biggest long value)
- 6) WAP to subtract two int values and see the result? Try different values.
- 7) WAP to multiply two int values and see the result? Then do this for doubles as well and check.
- 8) WAP to Divide two int values and get the output of remainder and quotient? Try to do this for double also.
- 9) Try to remember the precedence of each operator which is given below ( higher precedence value indicates higher priority):

ARITHMETIC OPERATORS :: Precedence : Associativity

a.) \* , / , % :: 2 : Left - Right

b.) + , - :: 1 : L - R

## RELATIONAL OPERATORS

c.) <, <=, >, >= :: 1 : L - R

d.) ==, != :: 2 : L - R

## LOGICAL OPERATORS

e.) ! (Logical Not) :: 1 : L - R

f.) && (Logical AND) :: 2

g.) || (Logical OR) :: 3

## BITWISE OPERATORS

h.) ~ (Bitwise NOT) :: 1 : L - R

i.) << (Left Shift) :: 2 : L - R

j.) >> (Right Shift) :: 2 : L - R

k.) & (Bitwise AND) :: 3 : L - R

l.) | (Bitwise OR) :: 4 : L - R

m.) ^ (Bitwise XOR) :: 5 : L - R

## CONDITIONAL OPERATOR

n.) ?: :: - : R - L

Try to come up with the result of the below SOPs on paper by applying the precedence given above and then execute the code and verify the o/p?

- a.) `System.out.println(1+2*5/2-2*4);`
- b.) `System.out.println(1*2*5-2+24);`
- c.) `System.out.println(1+2/5/2/2-4);`

10) Ternary operator syntax and working is

`(cond exp) ? x : y` => meaning if cond exp is true, x is returned else y.

Check the working of the following code:

`System.out.println(1==1 ? 5 : 6);`

`System.out.println(1==2 ? 5 : 6);`

`System.out.println(1==1 ? true : false);`

`System.out.println(1==2 ? false : true);`

Now execute the code given below and check the o/p ?

a.) `System.out.println( true ? false : true);`

b.) `System.out.println( true ? true: false);`

c.) `char x = 'X';`

`int i=0;`

`System.out.pritntln(true ? x : 0);`

`System.out.pritntln(false ? i : x);`

11) Execute the code given below and check the o/p ?

a.) `int i = 0;`

`boolean t = true;`

```
boolean f = false, b;  
b = (t & ((i++) == 0));  
b = (f & ((i+=2) > 0));  
System.out.println(i);
```

12) Execute the code given below and check the o/p ?

```
int i = 55;  
int j = 55;  
while(i <= j && j<=i & i==j){  
    System.out.println("Abc");  
}
```

### **Simple programs to help learn return / looping / passing parameters / switch / if / continue (IMPORTANT):**

0) Code a class MethodTest. Create main(). Create one more static method named public static void print(){SOP("in print()")}. Invoke print() method from main(). Put SOPs before invoking print() and after. Execute the class and verify the working of control flow. What SOPs are printed in what order? Do you understand this? Now put a return; statement in print() before SOP("in print()"); and then see if the compilation succeeds. Why not? Put the return statement after SOP as the last statement. Now compile and check. Why this difference in compiler output? Now change the return data type of print() to int. Compile the code without using any return statement in the method. Does it work? Why not? Now use return; as the last statement. Now compile and check again. Now code return 1; as the last statement. Do you understand that you have to return a value when you give a return data type in the method header? Now remove all the statements from print() and code this

```
public static int print()  
{
```

```
        if(1!=2)
            return 1;
    }
```

Check if it compiles. What if you add will this code compile? Ask questions to Lab Instructors if you do not understand the working of return.

1) Create a class Test. Create a method named process that accepts an int as a parameter. Create a main method. Create a int variable x with value 10. Print the value of x to monitor using SOP. Then invoke process() and pass x as a parameter. Print the value in process(). Change the value in process(). Print the value in main() after invoking process(). Are you understanding the control flow? Are you understanding how parameters are being passed?

2) Test main() calling another static method x() which should call another static method y() in the same class. Put SOPs in each method to understand flow of control. How does the JVM manage control flow? What data structure does it internally use? Create local variables in each method. Try to access one methods local variable in another method. Can you access it? Why not? Where are local variables stored in memory? How do you explain parameter passing? Now create another class with name Test2 and create a static method named z() in it. Call this method from Test's main() and see how control flow occurs. Create similar named local variables in main() and in x() and see if you get any errors during compilation. Why not? Talk to Lab Instructors if you cannot answer any of the questions given.

3) Accept an int as input to a method named printTables(int num) and then print the multiplication tables of the num from 1 to 10 to monitor (see end of document for solution code).

4) Accept 2 ints in a method, print the multiplication tables of the smaller number till the second number times (if 5,2 and given, print 2X1 till 2X5) => Print directly inside the method

5) Code a class named TestLoop with main(). Write the code to print the following when executed:

12345678910

12131415161718192021

2223242526272829303132

33...

Print all numbers in 1 line till you get a multiple of 11. Then skip the multiple of 11 and print the next set of numbers in next line.

6) Write a method to take an int number and int year as input and return number of days in that month as int. Which control statement will you use to effectively reduce writing more conditional code?

7) Print the following pattern in main() of a Test class using looping control statements (Think: 5 lines you need to print and in each line you need to print 1,2,3,4 and 5 values...how many loops do you need? See code solution given in the last part of the document if you do not get the answer to this)

1

12

123

1234

12345

8) Write a method to accept an int input. If the number is even, it should print "shambo" the given number of times. If it is odd, it should print "govinda" that many number of times.

9) Code a method to accept an int as input and print the following. If the int num is 4, printed output should be

1234

123

12

1

if the int num is 7, printed output should be

1234567

123456

12345

1234

123

12

1

10) Print this pattern to monitor in main()

```
      *
    *  *
  *  *  *
*  *  *  *
*  *  *  *  *
```

11) Print all character symbols to monitor from 1 to 65535. You should leave a space after each character symbol and if the numerical value is divisible by 5, you should skip to next number (which control statement will help you to go to end of iteration?).

**Code Solutions Hints (do not copy paste but trace the code and understand. Code it on your own without looking into it after):**

For 3rd problem:

```
public static void printTables(int num)
```



```
{  
    for(int i = 1 ; i <= 10 ; i++)  
        System.out.println("num"+"*" +i+"="+ (i*num));  
}
```

For 7th problem:

```
for(int k = 1 ; k <=5 ; k++) // for the 5 lines  
{  
    for (int j = 1 ; j <=k ; j++) // k number of times  
    {  
        System.out.print(j);  
    }  
    System.out.println(); // leave a new line  
}
```