



Uttara InfoSolutions

www.uttarainfo.com

Practicals 1:

Instructions:

- Install JDK, TextPad in your laptop by using help of Lab instructors. Set up the path environment variable in Windows (Control Panel -> System -> Adv Settings -> Environment variables -> System Env Variables -> select path -> click edit -> add ; to end and copy paste path till bin folder of JDK). Test whether javac.exe and java.exe work in command prompt (Start->cmd in windows).
- Create a directory (ideally in c:\) with pgms (or with your name, do not have space in the folder name please). Create 3 sub folders, source, classes and lib within it.
- Open two command prompts (windows start-> cmd two times).
In one, type in cd c:\pgms\source (enter) and cd c:\pgms\classes (enter).

Type in dir command and verify if the folders are empty.

Next fire TextPad / Notepad and type the following:

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
```

Save this as HelloWorld.java. Keep a track of cases of the letters.

In source command prompt, do the following:

```
c:\pgms\source> javac -d c:\pgms\classes HelloWorld.java
```

This should ideally compile. If it does not, understand why it is not compiling and ask the instructor to help you if you face a problem.

In the other command prompt, do the following:

```
c:\pgms\classes> java HelloWorld
```

Now do the following exercises:

- a) Create a Hello named .java file. Create a main() method in it. Use System.out.print("Hello") and System.out.println("HelloHai"); and see the difference. Understand the compilation and execution steps.
- b) Create a A.java. Create 3 class definitions named A, B, C. Try to make all of them be marked public. Have empty body. Compile A.java. Are you able to compile? If not why not? Mark B, C with package access (remove access specifier) and then compile. Now is it compiling? Why so? Insert main() in all 3 call definitions with different SOPs. Which main() will gets executed? Execute all the 3 class`s main().
- c) Try to change the header of the main() -> remove String[], change it to int[], make main() protected, etc and try to compile and run. What fails and why?
- d) Create an int variable in main() named i. Assign value 5 to it and print it out using SOP. Compile and test. Next do not assign any value to i and try to recompile. Do you get any error? why? what does it say?
- e) Create a byte variable in main(),

```
byte b = 100;
```

Try to assign 128 to a byte. Are you able to compile? what about -128? Try to compile, run.

- f) Create a byte variable, assign it a value and then store in an int variable.

```
byte b = 50;
int x = b; // why does this work?
b = x; // why does this not work?
```

- g) Replace last statement with b = (byte) x; and see if it works. Now test the same by assigning a value to a short, a char and try to assign to an int, long, float, double. Also from the bigger datatyped variables, put the value to smaller data typed variables. Do not create 2 variables with the same identifier.

- h) Try out all the arithmetic operators and print out the results.

```
int x = 10;
int y = 4;
```

```
int result1 = x + y;
int result2 = x * y;
int result3 = x / y;
int result4 = x % y;
int result5 = x - y;
```

```
System.out.println(result1);
System.out.println(result2);
System.out.println(result3);
```

```
System.out.println(result4);  
System.out.println(result5);
```

Now try to overflow and underflow the values in variables.

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

- a) `System.out.println("1"+2+3);`
- b) `System.out.println(6+2+"4");`
- c) `System.out.println('2'+2+"2");`
- d) `System.out.println('2'+2);`

Check on paper whether you are getting the correct result.

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

- a) `byte b = 10;`
`b = b + 10;`
`System.out.println(b);`

How do u fix the error for the above code.

- b) `byte b = 10;`
`b+=10;`
`System.out.println(b);`

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

- a) `char c = 'X';`
`int a = 63;`
`char d = a;`
`System.out.println(d);`
- b) `char c = 'X';`
`int a = 63;`
`char d = (char)a;`
`System.out.println(d);`

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

- a) `int a = 10/0; // you will have to comment this after testing`
- b) `double d = 10.0/0.0;`
- c) `float f = 10.0f/0.0f;`
- d) `double d = 10.0/0;`
- e) `double a = 10/0.0;`

m) Add 100 + 100 and store result in a byte. Print the value. Do you understand why is the result the one being printed?

Other programs (Do whichever ones you can. I will make you do the rest after the next 2 sessions):

- 1) Build a HelloWorld program
- 2) Print HelloWorld 10 times to the monitor
- 3) Print Hello with number (as Hello1, Hello2, Hello3...till Hello10) to the monitor
- 4) Print 1 to 10 to the monitor using for, while, do..while
- 5) Print all odd numbers from 1 to 100 to the monitor
- 6) Print all even numbers from 1 to 100 to the monitor
- 7) Print all non multiples of 3 between 1 to 100 to the monitor
- 8) Print "even" and the number or "odd" and the number between 1-100
- 9) Print "divisible" and the number if the num is divisible by 3 or 5 and not both between 1-100
- 10) Consider an int input (for ex 100). Using << and >>, multiply/divide the number by 2, 4, 8. Verify the results on paper.
- 11) WAP to Divide any two int values and get the output of remainder and quotient?
- 12) Create a byte variable with value 127. Add 10 to it and store in byte. Print the result. Is the result correct? Why not?
- 13) Execute the code given below and check the o/p ?

- a) `System.out.println(2 >> 1);`
- b) `System.out.println(6 >> 2);`
- c) `System.out.println(-2 >> 1);`

Check on paper whether you are getting the correct result.

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

- a) `System.out.println(10);`
- b) `System.out.println(012);`
- c) `System.out.println(0xAB);`

Check on paper whether you are getting the correct result.

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

- a) `System.out.println(2 << 1);`
- b) `System.out.println(6 << 2);`
- c) `System.out.println(-2 << 1);`

Check on paper whether you are getting the correct result.

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

- a) `System.out.println("1"+2+3);`
- b) `System.out.println(6+2+"4");`
- c) `System.out.println('2'+2+"2");`
- d) `System.out.println('2'+2);`

Check on paper whether you are getting the correct result.

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

- a) `int x = 10;`
`System.out.println(~x);`
- b) `int a = 100;`
`System.out.println(~a);`

Check on paper whether you are getting the correct result.

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

```
a) int x = 10;  
   int y = 6;  
   System.out.println(x | y);
```

```
b) int a = 100;  
   int b = 50;  
   System.out.println(a | b);
```

Check on paper whether you are getting the correct result.

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

```
a) int x = 10;  
   int y = 6;  
   System.out.println(x & y);
```

```
b) int a = 100;  
   int b = 50;  
   System.out.println(a & b);
```

Check on paper whether you are getting the correct result.

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

```
a) int x = 10;  
   int y = 6;  
   System.out.println(x ^ y);
```

```
b) int a = 100;  
   int b = 50;  
   System.out.println(a ^ b);
```

Check on paper whether you are getting the correct result.

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

```
a) int x = 10;  
   System.out.println(x >>> 2);
```

```
b) int a = -10;  
   System.out.println(a >>> 2);
```

Check on paper whether you are getting the correct result.

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

```
a) System.out.println(4 | 3);  
b) System.out.println(4 | 4);  
c) System.out.println(4 & 3);  
d) System.out.println(4 / 3);
```

Check on paper whether you are getting the correct result.

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

```
a) byte b = 10;
   b = b + 10;
   System.out.println(b);
```

How do u fix the error for the above code.

```
b) byte b = 10;
   b+=10;
   System.out.println(b);
```

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

```
a) char c = 'X';
   int a = 63;
   char d = a;
   System.out.println(d);
```

```
b) char c = 'X';
   int a = 63;
   char d = (char)a;
   System.out.println(d);
```

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

```
a) int a = 10/0; // you will have to comment this after testing
b) double d = 10.0/0.0;
c) float f = 10.0f/0.0f;
d) double d = 10.0/0;
e) double a = 10/0.0;
```

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

```
int i = 50;

while(i != 0){
    System.out.println(i >>>= 1);
}
```

27) 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?

28) Create multiple class definitions in 1 .java and see how many classes you can declare public. When you compile how many .class files are getting generated? Can multiple classes have main()? Which one will be executed by JVM, test it.

29) 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.

30) Put 'a' into a char. Put this variable into an int variable. Print out the value. Why do you the value you see being printed? Now print the next 25 values in a loop and print out the symbols by incrementing the char variable.

Sample code with syntax:

```
1)
public class HelloWorld
{
    public static void main(String[] args)
    {
        System.out.println("Hello World");
    }
}
2)
for(int i = 0 ; i < 10 ; i++)
{
}
3)
while(i > 10)
{
}
4)
do
{
}
while(i <=10);
5)
```

Use steps to program for these (code static methods):

- 1) Write a method to test whether an int input is even. Call this from main and test it. Test by using % operator and then by using bitwise operator.
- 2) WAP to take an input of int year and int month as parameter. Return the number of days in that month as an year.
- 3) WAP to test whether the int input is prime
- 4) WAP to test whether an int input contains an even sum (meaning if i/p is 12 -> 1+2 = 3 which is not even, if i/p is 123 -> 1 + 2 + 3 = 6 which is even)

i/o o/p
2 yes

3 no
12 no
123 yes
321 yes
234 no

```
public static boolean isSumEven(int num)
{
}
}
```

Ex:

```
public class Test
```

```
{
    public static boolean isEven(int num)
    {
        if(num<0)
            num = -num;

        return ((num& 1) == 0);
    }

    public static void main(String[] args)
    {
        System.out.println("is 1 even ?"+Hello.isEven(1));
        System.out.println("is 2 even ?"+Hello.isEven(2));
    }
}
```

5)WAP to convert an int input into binary representation and return it as a string.

6)WAP to test how many bits are set to 1 in a positive int variable.

7)WAP to test whether a passed int is a power of 2 (observe how many bits are set to 1 for a power of 2 and see if you can code for this)

8)WAP to add the alternate digits of a given input number and find their difference. Ex:
1234 -> (1+3) - (2+4) = -2, 123 -> 1+3 - 2 = 2.

9)WAP to have 2 methods to return the gcd and hcm of 2 ints passed as parameters.

10) WAP to accept an int and return if it is a palindrome.