EXPERIMENT 1 – INTRODUCTION TO JAVA

1. **Objectives:**

   (a). Learning Java Language
   (b). Difference between C++ and Java
   (c). Writing Programs in Java
   (d). Executing Java programs
   (e). Java variables
   (f). Arithmetic Expressions
   (g). Control Structures and Loops

2. **Time Required:** 6 hrs

3. **Programming Language:** Java

4. **Software Required:**
   (a). Windows OS
   (b). NetBeans / Eclipse Kepler

5. **Java Compiler** is software that compiles the Java code into Bytecode. Bytecode is a set of commands that the java virtual machine understands.

6. **Java Runtime Environment (JRE)** is the virtual machine over which the java code is executed. This virtual machine executes the java code. The JRE must be provided before executing a Java program.

7. **Java/NetBeans Download Links:**

   1. https://netbeans.org/downloads/
   2. https://www.oracle.com/java/technologies/javase-downloads.html

8. **Java naming rules and guidelines**

   An identifier is a word that can be used to name a class, method, variable, or constant. Java has rules for choosing a legal identifier. If a rule is not obeyed, the code will not compile. Recall that Java is case sensitive.

   Java Rules for Choosing Identifiers

   1. An identifier may use all letters, a – z and A – Z, all digits, 0 – 9, the underscore (_) and $.
   2. An identifier may not begin with a digit.
   3. An identifier may not be a **Java reserved word**. Reserved words, or key words are words that have a special meaning in Java.

Reserved words Examples:

abstact	boolean	break	byte	case	catch
char	class	const	continue	default	do
double	else	extends	final	finally	float
for	goto	if	implements	import	instanceof
int	interface	long	native	new	package
private	protected	public	return	short	static
strictfp	super	switch	synchronized	this	throw
throws	transient	try	void	volatile	while

Java Naming Guidelines

In addition to the naming rules, there are also guidelines, or conventions, that were established by the authors of the Java API. If a guideline is not obeyed, your code will compile, but it will be more difficult for you and others to read. Knowing these guidelines, will help you understand code that uses Java API classes.

1. **Identifiers:** Choosing names that indicate the purpose of the class, method or data value is known as **self – documentation.**
2. **Class Names:** begin with a capital letter, additional words are capitalized. Examples: FirstClass, HelloWorld, JFrame, JOptionPane.
3. **Variable Names:** begin with a lower case letter, additional words are capitalized. Examples: myWindow, visible, width, height, interestRate € Note: no parentheses
4. **Method Names:** begin with a lower case letter, additional words are capitalized. Examples: setTitle(),setSize(), showMessageDialog() € Note: always parentheses
5. **CONSTANTS:** are completely capitalized, additional words are separated with an underscore character _. Examples: PI,MAX_VALUE, INTEREST_RATE

At all times, you are expected to follow the Java guidelines for choosing identifiers. Failure to do so will result in points being deducted.

Syntax difference Between C++ and Java

C++ code	Java code

class Student
{
public:
string name;
int age;
string city;
};
Main
int main(void)
{
Student s1;
s1.name="Ali";
s1.age=26;
s1.city="Islamabad";
}
	class Student
	{
	Public string name;
	Public int age;
	Public String city;
	}
	Main
	public class myMain
	{
	public static void main(String[] args)
	{
	Student s1=new Student();
	s1.name="Ali";
	s1.age=26;
	s1.city="Islamabad";
	}
	}

10 **Program 01: " Printing Hello World "**

Class "MyFirstJavaProgram" has a function main. The main function calls a utility in System class that displays the string 'Hello World'

```
public class MyFirstJavaProgram {

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

Execute this code and show the Output:

11. **Variables** are constructs that allocate a memory location in RAM and place values in it. The capability of a variable is the ability to vary.

General format is:

```
type identifier [ = value][, identifier [= value] ...] ;
```

Data types in java are given here:
http://www.tutorialspoint.com/java/java_basic_datatypes.htm

```
int a, b, c;         // declares three ints, a, b, and c.
int d = 3, e, f = 5; // declares three more ints, initializing
                     // d and f.
byte z = 22;         // initializes z.
double pi = 3.14159; // declares an approximation of pi.
char x = 'x';        // the variable x has the value 'x'.
```

Program 03: Declaring and Initializing Variables in Java

```
public class EmployeeDetails {

public static void main(String args[])
        {

                int age = 0;
                age = age + 15;

                String firstName = "Ahmed";

                double salary = 15.4;

                System.out.println("Employee age is : " + age);
                System.out.println("Employee Name is : " + firstName);
                System.out.println("Employee Salary is : " + salary);
        }
}
```

Execute this code and show the Output:

12. **Program 03: Getting Input from User Using Scanner Class**

```
import java.util.*;
public class ScannerClassExample {
      public static void main(String args[])
{
          String s = "Hello, This is Scanner Class Example IET OOP Lab.";

          //Create scanner Object and pass string in it
          Scanner scan = new Scanner(s);
          //Check if the scanner has a token
          System.out.println("Boolean Result: " + scan.hasNext());
          //Print the string
          System.out.println("String: " +scan.nextLine());
          scan.close();
          System.out.println("--------Enter Your Details-------- ");

          Scanner in = new Scanner(System.in);
          System.out.print("Enter your name: ");
          String name = in.next();
          System.out.println("Name: " + name);
          System.out.print("Enter your age: ");
          int i = in.nextInt();
          System.out.println("Your Age: " + i);
          System.out.print("Enter your salary: ");
          double d = in.nextDouble();
          System.out.println("Salary: " + d);
          in.close();
          }
}
```

Execute this code and show the Output:

-

13. **<u>Program 04: Instance Variables Declaration/Initialization</u>**

```
import java.io.*;
class StudentMarks {
        // These variables are instance variables.
        // These variables are in a class
        // and are not inside any function
        int engMarks;
        int mathsMarks;
        int phyMarks;
}
class MarksDemo {
        public static void main(String args[])
        {
                // first object
                StudentMarks obj1 = new StudentMarks();
                obj1.engMarks = 20;
                obj1.mathsMarks = 80;
                obj1.phyMarks = 60;
                // second object
                StudentMarks obj2 = new StudentMarks();
                obj2.engMarks = 80;
                obj2.mathsMarks = 100;
                obj2.phyMarks = 15;
                // displaying marks for first object
                System.out.println("Marks for first object:");
                System.out.println(obj1.engMarks);
                System.out.println(obj1.mathsMarks);
                System.out.println(obj1.phyMarks);
                // displaying marks for second object
                System.out.println("Marks for second object:");
                System.out.println(obj2.engMarks);
                System.out.println(obj2.mathsMarks);
                System.out.println(obj2.phyMarks);
        }
}
```

Execute this code and show the Output:

14. **Program 05:** Write a Java program to display default value of all primitive data types of Java. (5)

Code:

Output:

15. **Program 06:** Write a Java program check if two strings are equal or not.
Code:

Output:

16. **Arithmetic Expressions** are a construct made up of variables and operators which are constructed according to the syntax of the language that evaluates to a single value.

int result = 1 + 2; // result is now 3

Operator	Description	Example

The expressions are same as in C/C++.

Java arithmetic operators are:

The Relational operators are:

The logical operators are: -

Program 07: Write java programs to give your own examples of each operator given below: **(Copied code will get you Zero Marks)**

1. Increment and decrement operators.
2. Bitwise Complement Operator.
3. Arithmetic operator.
4. Relational Operator
5. Bitwise operator.
6. Conditional Operator.

Code 01:

Output:

Code 02:

Output:

Code 03:

Output:

Code 04:

Output:

Code 05:

Output:

Code 06:

Output:

17. **Control Structures and Loops**

1. Control statements/Conditional Branches

These are used to choose the path for execution. There are some types of

control statements:

- If statement
- If-else statement
- If-else if statement/ ladder if statements
- Switch statement

2. **Loops in Java**

These are used to iterate the instruction for multiple times.

- For loop
- While loop
- Do-while loop

3. **Branching Statements in Java**

These are used to **alter the flow of control in loops.**

- Break
- Continue

Reference Material Sources:

- https://www.baeldung.com/java-control-structures#:~:text=There%20are%20three%20types%20in,flow%20of%20control%20in%20loops.
- https://javagoal.com/control-structures-in-java/

Program 08: IF-Else Example

```
public class Student {
public static void main(String[] args) {
int x = 10;
int y = 12;
if(x+y < 10) {
System.out.println("x + y is less than     10");
}  else {
System.out.println("x + y is greater than 20");
}
}
```

Execute and Show output of this Program:

—

Program 09: Nested If Statement

```
public class Student {
public static void main(String[] args) {
String address = "Delhi, India";

if(address.endsWith("India")) {
if(address.contains("Meerut")) {
System.out.println("Your city is meerut");
}else if(address.contains("Noida")) {
System.out.println("Your city is noida");
}else {
System.out.println(address.split(",")[0]);
}
}else {
System.out.println("You are not living in india");
}
}
}
```

Execute and Show output of this Program:

Program 10: Do-While Loop

```
public class Calculattion {
public static void main(String[] args) {

int i = 0;
System.out.println("Printing the list of first 20 even numbers \n");
do {
System.out.println(i);
i = i + 4;
}while(i<=20);
}
}
```

Execute and Show output of this Program:

Program 11: Break Statement using For Loop

```
public class BreakExample {

public static void main(String[] args) {
// Break statement example using loop

for(int i = 5; i<= 20; i++)

 {
System.out.println(i);
if(i==15) {
break;
}
}
}
```

Execute and Show output of this Program:

Exercise:

1. Write a java program to find the Cube of an integer number.
2. Write a java program to display the first 20 odd numbers.
3. Write a java Program to generate a Ladder of numbers.
4. Write a program to give your own examples of control statements.

    1. Nested for loop.
    2. Nested If-else Statements
    3. Do statements

Web Resources

https://netbeans.org/downloads/
http://www.oracle.com/technetwork/java/javase/downloads/jdk-7-netbeans-download-432126.html
http://www.learnjavaonline.org/
http://docs.oracle.com/javase/tutorial/

Summary: This is the basic tutorial of Java that allows the student to create basic Java programs similar to C++/C. The programs include variables, arithmetic statements and Operators , Control Structures – Loops.