CS 8392 & OBJECT ORIENTED PROGRAMMING

UNIT I

INTRODUCTION TO OOP AND JAVA FUNDAMENTALS

Topic 11 : Control Statements and Arrays

Topic 12 : Packages

KNOWLEDGE INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to Anna University, Accredited by NAAC and NBA (B.E : Mech., ECE, EEE & CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course code & Name : CS 8392 & Object Oriented Programming

Year / Semester : II / III

Academic Year : 2020 – 2021

Presented By : Mrs.C.Vanitha, AP/CSE,KIOT

CONTROL STATEMENTS

A programming language uses control statements to control the flow of execution of program based on certain conditions.

SELECTION STATEMENTS:

- if
- if-else
- nested-if
- if-else-if
- switch-case
- jump – break, continue, return

ITERATIVE STATEMENTS

In programming languages, loops are used to execute a set of instructions/functions repeatedly when some conditions become true. There are three types of loops in java.

- while loop
- do-while loop
- For loop

JUMP STATEMENTS:

- break
- Continue

if statement

if statement tests the condition. It executes the *if block* if condition is true.

Syntax

```
if(condition){
//code to be executed
}
```

CONTROL STATEMENTS

Example:

```
public class IfExample {
public static void main(String[] args) {
       int age=20;
        if(age>18){
  System.out.print("Age is greater than 18");

   }
}
}
```

Output:

Age is greater than 18

If-else statement

if-else statement also tests the condition. It executes the *if block* if condition is true otherwise *else block* is executed.

Syntax:

```
if(condition){
//code if condition is true
}else{
//code if condition is false
}
```

Example:

```
public class IfElseExample {
public static void main(String[] args) {
     int number=13;
     if(number%2==0){
       System.out.println("even number");
   }
else
{
       System.out.println("odd number");
   }
}
}
```

Output:

odd number

CONTROL STATEMENTS

Java Nested if statement

•The nested if statement represents the *if block within another if block*. Here, the inner if block condition executes only when outer if block condition is true.

Syntax:

```
if(condition){
    //code to be executed
        if(condition){
          //code to be executed
   }
}
public class nestif {
public static void main(String[] args) {
      int age=20;
   int weight=80;
       if(age>=18){
       if(weight>50)
{ System.out.println("You are eligible to don
ate blood");
       }      }  }}
```

Switch statement:

Executes one statement from multiple conditions

It is like if else if ladder statement.

Syntax:

```
switch(expression){
case value1:
 //code to be executed;
 break;  //optional
case value2:
 //code to be executed;
 break;  //optional
.....

default:
// code to be executed if all cases are not
matched;
}
```

CONTROL STATEMENTS

Example:

```
public class SwitchExample {
public static void main(String[] args) {
      int number=20;

   switch(number){

   case 10: System.out.println("10");
   break;
   case 20: System.out.println("20");
   break;
   case 30: System.out.println("30");
   break;
   default:
System.out.println("Not in 10, 20 or 30");
   }
}
}
```

ITERATIVE STATEMENTS

for loop is used to iterate a part of the program several times. If the number of iteration is fixed, it is recommended to use for loop.

```
for(initialization;condition;incr/decr
)
{
//statement or code to be executed
}

public class ForExample {
public static void main(String[] args) {
   //Code of Java for loop
   for(int i=1;i<=10;i++){
      System.out.println(i);
   }
}
}
```

CONTROL STATEMENTS

while loop is used to iterate a part of the program several times. If the number of iteration is not fixed, it is recommended to use while loop.

Syntax:

```
while(condition){
//code to be executed
}
```

```
public class WhileExample {
public static void main(String[] args) {
    int i=1;
    while(i<=10){
        System.out.println(i);
    i++;
    }
}
}
```

do-while loop is used to iterate a part of the program several times.

The Java ***do-while loop*** is executed at least once because condition is checked after loop body.

Syntax:

```
do{
//code to be executed
}while(condition);
```

```
public class DoWhileExample {
public static void main(String[] args) {
    int i=1;
    do{
        System.out.println(i);
    i++;
    }while(i<=10);
}
}
```

CONTROL STATEMENTS

JUMP STATEMENTS

break statement is used to break loop or switch statement. It breaks the current flow of the program at specified condition. In case of inner loop, it breaks only inner loop.

The loop is immediately terminated and the program control resumes at the next statement following the loop.

Syntax:

break;

```
        class BreakExample {
public static void main(String[] args) {
   //using for loop
   for(int i=1;i<=10;i++){
      if(i==5){
         //breaking the loop
         break;
      }
      System.out.println(i);
   } } }
```

continue statement is used in loop control structure when you need to jump to the next iteration of the loop immediately. It can be used with for loop or while loop.

It continues the current flow of the program and skips the remaining code at the specified condition

Syntax:

```
continue;
public class ContinueExample {
public static void main(String[] args) {
      for(int i=1;i<=10;i++){
      if(i==5){
         continue;
      }
      System.out.println(i);
   } }
```

ARRAYS

An array is a collection of similar type of elements. The elements of an array are stored in a contiguous memory location.

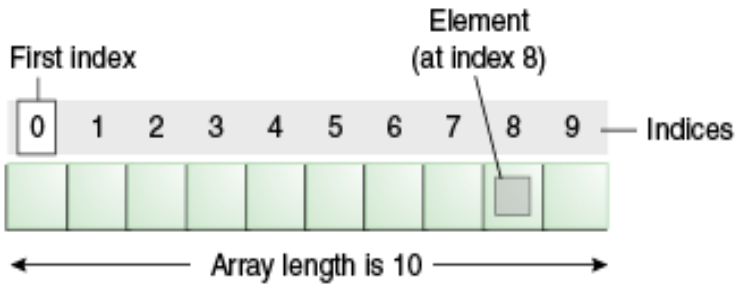
•Array in Java is index-based, the first element of the array is stored at the 0th index, 2nd element is stored on 1st index and so on.

•In Java, array is an object of a dynamically generated class.

Two types:

•Single Dimensional Array

•Multidimensional Array



Syntax:

dataType[] arr; (or)
dataType []arr; (or)
dataType arr[];

Instantiation of array

arrayRefVar=**new** datatype[size];

```
class Testarray{
public static void main(String args[]){
int a[]=new int[5];//declaration and instantiation
a[0]=10;//initialization
a[1]=20;
a[2]=70;
a[3]=40;
a[4]=50;
//traversing array
for(int i=0;i<a.length;i++)
System.out.println(a[i]);
}}
```

ARRAYS

Multi dimentional arrays:

Data is stored in row and column based index (also known as matrix form).

Syntax:

```
int[][] arr=new int[3][3];
```

```
class Testarray3{
public static void main(String args[]){
int arr[][]={{1,2,3},{2,4,5},{4,4,5}};
//printing 2D array
for(int i=0;i<3;i++){
 for(int j=0;j<3;j++){
   System.out.print(arr[i][j]+" ");
 }
 System.out.println();
}
}}
```

Output:

```
1 2 3
2 4 5
 4 4 5
```

Packages

•A java package is a group of similar types of classes, interfaces and sub-packages. Package in java can be categorized in two form, built-in package and user-defined package. There are many built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.

Advantages:

1. Used to categorize the classes and interfaces so that they can be easily maintained.
2. Provides access protection
3. Removes naming collision.

Defining a Package:

package <fully qualified package name>;

package *pkg*;

pkg is the name of the package

The general form of a multileveled package statement is

package pkg1[.pkg2[.pkg3]];

A package hierarchy must be reflected in the file system

Syntax

```
package java.awt.image;
```

Example

```
package pack;
public class A{
  public void msg()
{
System.out.println("Hello");
}
}

package mypack;
import pack.*;
  class B{
  public static void main(String args[]){
   A obj = new A();
   obj.msg();
  }
}
```

Output:

Hello

Packages

To compile : javac –d . B.java
To Run: java mypack.B
Example:

```
package pck1;
class Student
{
private int rollno;
private String name;
private String address;
public Student(int rno, String sname, String sadd){
rollno = rno;
name = sname;
address = sadd;
}
public void showDetails()
{
System.out.println("Roll No :: " + rollno);
System.out.println("Name :: " + name);
System.out.println("Address :: " + address);
}}
```

```
public class DemoPackage
{
public static void main(String ar[])
{
Student st[]=new Student[2];
st[0] = new Student (1001,"Alice", "New
York");
st[1] = new
Student(1002,"BOb","Washington");
st[0].showDetails();
st[1].showDetails();
}
}
```

Output:

```
Roll No :: 1001
Name :: Alice
Address :: New York
Roll No :: 1002
Name :: Bob
Address :: Washington
```

REFERENCES

- Herbert Schildt, “Java The complete reference”, 8th Edition, McGraw Hill Education, 2011.

- Cay S. Horstmann, Gary cornell, “Core Java Volume –I Fundamental”, 9th Edition, Prentice Hall, 2013.

- www.javatpoint.com

THANK YOU