Chapter 7

Program 1 if condition

```
public class IF
{
  public static void main(String args[])
{
      int num=70;
          if( num < 100 )
{
        System.out.println("number is less than 100");
        if(num > 50)
{
              System.out.println("number is greater than 50");
            }
          }
  }
}
```

Program 2 for loop

```
class FOR
{
   public static void main(String args[])
{
       for(int i=10; i>1; i--)
{
          System.out.println("The value of i is: "+i);
      }
   }
}
```

Program 3 java program for pyramid

```
public class Pyramid1
{
      public static void main(String[] args)
{
           for(int i=1; i<= 5 ;i++)
{
                  for(int j=0; j < i; j++)
{
                      System.out.print("*");
                }
                   System.out.println("");
           }
     }
}
```

Output of the above program would be

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

Program 4 Mathematical operators

```
public class Operators
 {
      public static void main(String[] args)
{
  System.out.println("Arithmetic operators example :");
           int i = 50 + 20;
           int j = i - 10;
           int k = j * 2;
```

```
        double l = k / 6;
        System.out.println("i = " + i);
        System.out.println("j = " + j);
        System.out.println("k = " + k);
        System.out.println("l = " + l);
    }
}
```

Output would be

```
Arithmetic operators example:
i = 70
j = 60
k = 120
l = 20.0
```

Program 5 Auto-increment and Auto-decrement Operators

```
public class AutoOperatorDemo
{
  public static void main(String args[])
{
    int num1=100;
    int num2=200;
    num1++;
    num2--;
    System.out.println("num1++ is: "+num1);
    System.out.println("num2-- is: "+num2);
  }
}
```

Output:

```
num1++ is: 101
num2-- is: 199
```

Program 6 Switch Case

```
public class SwitchCase
 {
  public static void main(String args[])
{
    int i=2;
    switch(i)
    {
          case 1:
            System.out.println("Case1 ");
            break;
          case 2:
            System.out.println("Case2 ");
            break;
          case 3:
            System.out.println("Case3 ");
            break;
          case 4:
       System.out.println("Case4 ");
       break;
          default:
            System.out.println("Default ");
    }
  }
}
```

Program 7 For loop

```
class ForLoop
{
```

```
   public static void main(String args[])
{
      for(int i=10; i>1; i--){
         System.out.println("The value of i is: "+i);
      }
   }
}
```

The output of this program is:
The value of i is: 10
The value of i is: 9
The value of i is: 8
The value of i is: 7
The value of i is: 6
The value of i is: 5
The value of i is: 4
The value of i is: 3
The value of i is: 2

Program 8 Whileloop

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

Output:
10
9
8
7
6
5
4
3
2

Program 9 Dowhile

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

Output:
10
9
8
7
6
5

4
3
2

Program 10 Java Nested Loop

```
class NestedLoop
 {
  public static void main(String[] args)
 {
    int i = 1;

     while (i <= 5)
{
    System.out.println("Outer loop iteration " + i);
    for (int j = 1; j <= 2; ++j) {
         System.out.println("i = " + i + "; j = " + j);
       }
        ++i;
    }
  }
}
```

The output of this program and above program is same.

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

Program 12 Labelled loop

```
   class  LabelledLoop
  {
     public static void main(String args[])
     {
        int i,j;

        loop1:   for(i=1;i<=10;i++)
        {
           System.out.println();

           loop2:   for(j=1;j<=10;j++)
           {
              System.out.print(j + " ");
              if(j==5)
                 break loop1;
           }
        }
     }
  }
```

Program 13 sort hand operators

```
public class sorthand
{
   public static void main(String[] args)
 {
            int i = 5;
             int j = 10;
             i += 5;
             j -= 2;
             System.out.println("i = " + i);
             System.out.println("j = " + j);
               }
}
```

Output would be
i = 10
j = 8

Program 13 modulus operator

```
public class Modulus
{
    public static void main(String[] args)
{
   System.out.println("Java Modulus Operator example");
   int i = 50;
  double d = 32;

   System.out.println("i mod 10 = " + i%10);
   System.out.println("d mod 10 = " + d%10);
      }
}
```

Output would be

Java Modulus Operator example
i mod 10 = 0
d mod 10 = 2.0
program 21:

```
class ConditionalDemo2
{
   public static void main(String[] args)
{
      int value1 = 1;
      int value2 = 2;
      int result;
      boolean someCondition = true;
      result = someCondition ? value1 : value2;
      System.out.println(result);
  } }
```

Chapter 8

Program : 1 min function

```
public class MinNumber {
 public static void main(String[] args)
{
     int a = 11;
     int b = 6;
     int c = minFunction(a, b);
     System.out.println("Minimum Value = " + c);
  }
  /** returns the minimum of two numbers */
  public static int minFunction(int n1, int n2)
{
     int min;
     if (n1 > n2)
       min = n2;
     else
       min = n1;
     return min;
  }
}
```

Output

Minimum value = 6

Program 2: polymorphism

```
public class Employee
 {
  private String name;
  private String address;
```

```
  private int number;
  public Employee(String name, String address, int number)
 {
    System.out.println("Constructing an Employee");
    this.name = name;
    this.address = address;
    this.number = number;
  }
  public void mailCheck()
 {
    System.out.println("Mailing a check to " + this.name + " " + this.address);
  }
  public String toString()
 {
    return name + " " + address + " " + number;
  }

  public String getName()
{
    return name;
  }
  public String getAddress()
{
    return address;
  }
  public void setAddress(String newAddress)
{
    address = newAddress;
  }
  public int getNumber()
{
    return number;
  }
}
```

Program 3: Example of Constructor Overloading

```
class Student5
{
   int id;
   String name;
   int age;
   Student5(int i,String n)
{
   id = i;
   name = n;
   }
   Student5(int i,String n,int a)
{
   id = i;
   name = n;
   age=a;
   }
   void display(){System.out.println(id+" "+name+" "+age);
}
   public static void main(String args[])
{
   Student5 s1 = new Student5(111,"Karan");
   Student5 s2 = new Student5(222,"Aryan",25);
   s1.display();
   s2.display();
  }
```

```
}
```

Program 4. Private modifier

```
class A
{
private int data=40;
private void msg(){System.out.println("Hello java");
}
}
public class Simple{
 public static void main(String args[]){
   A obj=new A();
   System.out.println(obj.data);//Compile Time Error
   obj.msg();//Compile Time Error
   }
}
```

Program 5 Package

```
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

6 Program Single Inheritance Example

File: TestInheritance.java

```
class Animal
{
void eat(){System.out.println("eating...");
}
}
class Dog extends Animal
{
void bark(){System.out.println("barking...");
}
}
class TestInheritance{
public static void main(String args[])
{
Dog d=new Dog();
d.bark();
d.eat();
}
}
```

Chapter 9

Program 1: 1D array

```
lass 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;
//printing array
for(int i=0;i<a.length;i++)//length is the property of array
System.out.println(a[i]);
}
}
```

Output: 10
20
70
40
50

Program 2 2D array

```
class Testarray3
{
public static void main(String args[])
{
 //declaring and initializing 2D array
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

Program 3 upper case to lower case

```
public class Case
 {
 public static void main(String[] args)
{
 String str = "STRING TOLOWERCASE EXAMPLE";
 String strLower = str.toLowerCase();
 System.out.println("Original String: " + str);
 System.out.println("String changed to lower case: " + strLower);
      }
}
```

Output
Original String: STRING TOLOWERCASE EXAMPLE
String changed to lower case: string to lower case example

Program 4 Java String concat() method example

```
public class Concat
{
public static void main(String args[])
{
String s1="java string";
s1.concat("is immutable");
```

```
System.out.println(s1);
s1=s1.concat(" is immutable so assign it explicitly");
System.out.println(s1);
}
}
```

Output

```
java string
java string is immutable so assign it explicitly
```

program 5 Java String length() method example

```
public class Length
{
public static void main(String args[])
{
String s1="javatpoint";
String s2="python";
System.out.println("string length is: "+s1.length());//10 is the length of javatpoint string
System.out.println("string length is: "+s2.length());//6 is the length of python string
}
}
```

output

```
string length is: 10
string length is: 6
```

Program 6 Java String split() method example

```
//The given example returns total number of words in a string excluding space only. It also includes special characters.
public class Split
{
public static void main(String args[])
{
String s1="java string split method by javatpoint";
String[] words=s1.split("\\s");//splits the string based on whitespace
//using java foreach loop to print elements of string array
for(String w:words)
{
System.out.println(w);
}
}}
```

Output

```
java
string
split
method
by
javatpoint
```

Program 7 reverse string

```
 public class reverse
{
public static void main(String[] args)
{
   System.out.println(StringFormatter.reverseString("my name is khan"));
   System.out.println(StringFormatter.reverseString("I am sonoo jaiswal"));
   }
}
```

Output:

```
nahk si eman ym
lawsiaj oonos ma I
```

program 8 Java Calendar Class Example: get()

```
import java.util.*;
public class CalendarExample2
{
```

```
 public static void main(String[] args)
{
  Calendar calendar = Calendar.getInstance();
  System.out.println("At present Calendar's Year: " + calendar.get(Calendar.YEAR));
  System.out.println("At present Calendar's Day: " + calendar.get(Calendar.DATE));
  }
}
```

Output:

```
At present Calendar's Year: 2017
At present Calendar's Day: 20
```

Program 9 Date class

```
import java.util.Date;
public class DateDemo
{
  public static void main(String args[])
{
    // Instantiate a Date object
    Date date = new Date();
    // display time and date using toString()
    System.out.println(date.toString());
  }
}
```

Output

```
on May 04 09:51:52 CDT 2009
```

Chapter 10

Programs 1 Exception handler

```
public class Test
{
   public static void main(String[] args)
   {
      try
      {
         System.out.printf("1");
         int data = 5 / 0;
      }
      catch(ArithmeticException e)
      {
         System.out.printf("2");
         System.exit(0);
      }
      finally
      {
         System.out.printf("3");
      }
      System.out.printf("4");
   }
}
```

Chapter 11.

Program 1

```
import java.io.*;
public class FileDemo
{
   public static void main(String[] args)
{
       try
{
         File file = new File("javaFile123.txt");
         if (file.createNewFile())
{
            System.out.println("New File is created!");
```

```
        }
else {
            System.out.println("File already exists.");
        }
      } catch (IOException e)
 {
        e.printStackTrace();
      }

   }
}
```

Output:

```
New File is created!
```

Program 2 copy file

```
import java.io.*;
public class CopyFile
{
  public static void main(String args[]) throws IOException
{
    FileInputStream in = null;
    FileOutputStream out = null;
    try
{
      in = new FileInputStream("input.txt");
      out = new FileOutputStream("output.txt");
          int c;
      while ((c = in.read()) != -1)
{
        out.write(c);
      }
    }
Finally
 {
      if (in != null)
 {
        in.close();
      }
      if (out != null)
 {
        out.close();
      }
    }
  }
}
//Now let's have a file input.txt with the following content –This is test for copy file.
```

Program 3 reader and writer class

```
import java.io.*;
public class CopyFile
 {
  public static void main(String args[]) throws IOException
 {
    FileReader in = null;
    FileWriter out = null;
    try
{
      in = new FileReader("input.txt");
      out = new FileWriter("output.txt");
     int c;
      while ((c = in.read()) != -1)
{
```

```
        out.write(c);
      }
    }
Finally
{
      if (in != null)
{
        in.close();
      }
      if (out != null)
{
        out.close();
      }
    }
  }
}
```

//Now let's have a file input.txt with the following content –This is test for copy file.

Pro 4 console

```
import java.io.*;
public class ReadConsole
{
 public static void main(String args[]) throws IOException
{
    InputStreamReader cin = null;
    Try
{
      cin = new InputStreamReader(System.in);
      System.out.println("Enter characters, 'q' to quit.");
      char c;
      do
{
        c = (char) cin.read();
        System.out.print(c);
      }
 while(c != 'q');
    }
Finally
{
      if (cin != null)
{
        cin.close();
      }
    }
  }
}
```

Output

Let's keep the above code in ReadConsole.java file and try to compile and execute it as shown in the following program. This program continues to read and output the same character until we press 'q' –

```
$javac ReadConsole.java
$java ReadConsole
Enter characters, 'q' to quit.
1
1
e
e
q
q
```

Program 5 file I/O stream

```
import java.io.*;
public class fileStreamTest
{
  public static void main(String args[])
 {
       Try
 {
       byte bWrite [] = {11,21,3,40,5};
       OutputStream os = new FileOutputStream("test.txt");
       for(int x = 0; x < bWrite.length ; x++)
 {
         os.write( bWrite[x] );   // writes the bytes
       }
       os.close();
          InputStream is = new FileInputStream("test.txt");
       int size = is.available();

       for(int i = 0; i < size; i++)
 {
         System.out.print((char)is.read() + "  ");
       }
       is.close();
    } catch (IOException e)
 {
       System.out.print("Exception");
    }
  }
}
```

// The above code would create file test.txt and would write given numbers in binary format. Same would be the output on the stdout screen.

Program 6 Scanner class

```
import java.util.Scanner;
class ScannerTest{
 public static void main(String args[])
{
   Scanner sc=new Scanner(System.in);
     System.out.println("Enter your rollno");
   int rollno=sc.nextInt();
   System.out.println("Enter your name");
   String name=sc.next();
   System.out.println("Enter your fee");
   double fee=sc.nextDouble();
```

```
  System.out.println("Rollno:"+rollno+" name:"+name+" fee:"+fee);
  sc.close();
 }
}
```

Output:

```
Enter your rollno
111
Enter your name
Ratan
Enter
450000
Rollno:111 name:Ratan fee:450000
```