

MODULE 2 - SHEET 1

```
public class ComeIn
{ public static void main(String[] args)
  { System.out.println("Come in number 57 please");
  }
}
```

```
public class ComeAgain
{ private static int n;

  public static void main(String[] args)
  { n = 57;
    System.out.println("Come in number " + n + " please");
  }
}
```

```
public class ComeFinal
{ public static void main(String[] args)
  { int n;
    n = 57;
    System.out.println("Come in number " + n + " please");
  }
}
```

```
public class TwiceA
{ public static void main(String[] args)
  { int i, j;
    i = 57;
    j = 2*i;
    System.out.println("Twice " + i + " is " + j);
  }
}
```

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

```

    { int i, j;
      i = 57;
      j = twice(i);
      System.out.println("Twice " + i + " is " + j);
    }

private static int twice(int m)
{ int n;
  n = 2*m;
  return n;
}
}

public class TwiceC
{ public static void main(String[] args)
  { int i = 57;
    System.out.println("Twice " + i + " is " + twice(i));
  }

  private static int twice(int k)
  { return 2*k;
  }
}

public class ArgsArrayA
{ public static void main(String[] args)
  { System.out.println("There are " + args.length + " arguments");
    System.out.println(args[0] + " " + args[1] + " " + args[2]);
  }
}

// When run as  java ArgsArrayA Tom Dick Harry  this yields:
//
// There are 3 arguments
// Tom Dick Harry
//
// When run as  java ArgsArrayA  this yields:
//
// There are 0 arguments
//
// and then throws an  ArrayIndexOutOfBoundsException

```

MODULE 2 - SHEET 2

Some Java Constructs

Loops:

```
while (<condition>)
{ <statement>
  <statement>
  <statement>
}

do
{ <statement>
  <statement>
  <statement>
} while (<condition>);

while (true)
{ <statement>
  <statement>
  <statement>
}

for (<initialize>; <check>; <update>)
{ <statement>
  <statement>
  <statement>
}
```

If Statements:

```
if (<condition>) <statement>

if (<condition>)
{ <statement>
  <statement>
  <statement>
}

if (<condition>)
{ <statement>
  <statement>
  <statement>
}
else
{ <statement>
  <statement>
  <statement>
}
}
```

Ternary if-else operator:

<condition> ? <true-value> : <false-value>

Relational Operators:

< <= > >= == !=

Reserved Words:

abstract	default	goto	operator	synchronized
boolean	do	if	outer	this
break	double	implements	package	throw
byte	else	import	private	throws
byvalue	extends	inner	protected	transient
case	false	instanceof	public	true
cast	final	int	rest	try

MODULE 2 - SHEET 3

```
public class SumA
{ public static void main(String[] args)
  { int sum=0, i=1;
    while (i<=100)
{ sum = sum+i;
    i = i + 1;
    }
    System.out.println("Sum is " + sum);
  }
}
```

```
public class SumB
{ private static int n=100;

  public static void main(String[] args)
  { int sum=0, i=1;
    while (i<=n)
{ sum = sum+i;
    i = i + 1;
    }
    System.out.println("Sum is " + sum);
  }
}
```

```
public class SumC
{ private static int n=100;

  public static void main(String[] args)
```

```

    { int sum=0, i=1;
      do
      { sum += i;
        i += 1;
      } while (i<=n);
      System.out.println("Sum is " + sum);
    }
  }
}

```

```

public class SumD
{ private static int n=100;

  public static void main(String[] args)
  { int sum=0, i=1;
    while (true)
  { sum += i;
    if (i==n)
    { break;
    }
    i++;
  }
  System.out.println("Sum is " + sum);
}
}

```

```

public class SumE
{ private static final int N=100;

  public static void main(String[] args)
  { int sum=0;
    for (int i=1; i<=N; i++)
    { sum += i;
    }
    System.out.println("Sum is " + sum);
  }
}

```

```

public class ArgsArrayB
{ public static void main(String[] args)
  { System.out.println("There are " + args.length + " arguments");
    for (int i=0; i<args.length; i++)

```

```

        System.out.print(args[i] + " ");
        System.out.println();
    }
}

```

MODULE 2 - SHEET 4

```

public class ArrayIntro
{
    public static void main(String[] args)
    {
        int k = 1;
        int[] jack;                // To save a line write:
        jack = new int[4];          // int[] jack = new int[4];
        jack[1] = 10;
        jack[3] = jack[1] - 5;
        jack[0] = jack[3];
        jack[k+1] = 6;
        int sum = 0;
        for (int i=0; i<4; i++)
            sum += jack[i];
        System.out.println("Total is " + sum);

        int[] jill = {3,1,4,1,5,9};
        System.out.println("Length is " + jill.length);
        sum = 0;
        for (int i=0; i<jill.length; i++)
            sum += jill[i];
        System.out.println("Total is " + sum);

        jack = jill;
    }
}

```

```
public class SumArr
{ public static void main(String[] args)
  { int[] tom = {6,2,1,4,3,5};
    System.out.println("Total of tom is " + sum(tom));
  }

  private static int sum(int[] a)
  { int s = 0;
    for (int i=0; i<a.length; i++)
      s += a[i];
    return s;
  }
}
```