JAVA SYNTAX

JAVA Variables

Datatype variablename;

Example:

int a;

String b;

float c;

double d;

char e;

Syntax for importing a package:

import packagename;

Example for importing a package:

```
import java.util.*;
import java.io.*;
import java.util.regex*;
import java.nio.*;
```

Syntax for creating a class

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

    }
}
```

Syntax for creating an object

classname objname=new classname();

Decision Making Statements

If Condition

- Simple if
- If-Else
- If-Else Ladder
- Nested If

Switch Statement

If Condition Syntax:

```
if(condition)
{
        //Statements to be executed;
}
```

Example for Simple if Condition

```
if(a>0)
{
        System.out.println("Positive Number");
}
```

If-Else Statement Syntax:

```
if(condition)
{
        //Statements to be executed
}
else
{
        //statements to be executed
}
```

Example for if-else statement

```
if(a>0)
{
        System.out.println("Positive Number");
}
else
{
        System.out.println("Negative Number");

}
```

Else if ladder Statement Syntax:

```
if(condition)
{
        //statements to be executed
```

```
}
else if(condition)
{
        //statements to be executed
}
else if(condition)
{
        //statements to be executed
}
.
.
.
.
else
{
        //statements to be executed
}
```

Example for else if ladder statement:

```
if(a>b)
{
        //statements to be executed
}
else if(b>c)
{
        //statements to be executed
}
else if(a>c)
{
        //statements to be executed
}
else
{
```

```
        //statements to be executed
    }
```

Nested If Statement Syntax

```
    if(condition)
    {
        if(condition)
        {

        }
    }
```

Example for Nested if statement

```
    if(a>0)
    {
        if(a%2==0)
        {

        }
    }
```

Switch Statement Syntax

```
    switch(expression)
    {
        case 1:
            //statement to be executed;
            break;
        case 2:
            //statement to be executed;
            break;
        case 3:
            //statement to be executed;
            break;
        .
.       .
        .
```

```
        .
        .
        default:
                //statement to be executed;
}
```

Example for Switch statement

```
switch(a)
{
        case 1:
                //statement to be executed;
                break;
        case 2:
                //statement to be executed;
                break;
        case 3:
                //statement to be executed;
                break;
        .
.       .
        .
        .
        .
        default:
                //statement to be executed;
}
```

Looping Statement

Do-While Loop

While Loop

For loop

Do-While Loop syntax

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

Do-While Loop Example

```
do
{
        //statements to be executed;
}while(a!=0)
```

While Loop syntax

```
while
{
        //statements to be executed;
}
```

While Loop Example

```
while(a!=0
{
        //statements to be executed;
}
```

For Loop Syntax

```
for(initialization;testcondition;increment/decrement)
{


}
```

For Loop Example:

```
for(i=0;i<10;i++)
{
        //statements to be executed;
```

```
}
```

Syntax for creating functions

```
returntype functionname()
{
        //statements to be executed
}
```

Example for creating a function

```
void getdata()
{
        //statements to be executed;
}
```

Syntax for creating constructor

```
classname()
{
        //statements to be executed
}
```

Example for creating constructor

```
class Student
{
        Student()
        {
                //statements to be executed;
        }
}
```

Syntax for deriving a class

```
class derivedclassname extends baseclassname
{
}
```

Example for deriving a class

```
class Student
{
```

```
}
class Exam extends Student
{

}
```

Syntax for Overloading a function

```
void function_name()
{
}
void function_name(arguments)
{
}
```

Example1 for Overloading a function

```
void getdata()
{
}
void getdata(int a)
{
}
```

Example2 for Overloading a function

```
void getdata()
{
}
int getdata()
{
}
```

Example3 for Overloading a function

```
void getdata()
{
}
```

```
        void getdata(int a)
        {
        }
```

Syntax for creating a generic class

```
Class Name_Of_The_Class<GenericType>
{
}
```

Example for creating a generic class

```
Class Student<T>
{

}
```

Syntax for creating a static variable

```
static datatype variablename;
```

Example for creating a static variable

```
static int a;
```

Syntax for creating a static class

```
static class Name_Of_The_Class
{

}
```

Exxample for creating a static class

```
static class Student
{

}
```