Java Mastery - Part 2 Cheat Sheet by Bayan (Bayan.A) via cheatography.com/122738/cs/22936/

if then else	Calling a Method	Return variable from method to main: (cont)
if statement identifies the statement/code	When the method is called, all the code in curl	ly br Addste{ bioghænmæthiaddossæxæøstter accounted
block to run based on the expressions	Execute by calling the method (in main) with t	the rEqllies:changlements in the brackets
value(the specific condition)	Arguments can be the variable name or the ac	ctuaRValles wetatic officient (S tring[]
Code block is {can contain 1 or	want to send, as long as it matches the param	eters error.
defined by curly more statements}	method Nam e(v ari able1, variab l	.e2); boolean isThis Helpful = true;
braces	In the example, two arguments expected	int points = 15; datatype need
else comes after the if, executed when the		datatype need method Nam e(i sTh isH elpful, p not be specified
condition is false	A variable can be assigned a method result	}
else if can test multiple conditions	e.g.	public static int method Nam e(b
Example	<pre>int highScore = calcul ate Sco re(nus);</pre>	oolean isThis Hel pful, int poi gam eOver, score, level, bo nts) {
int score = 6000;	When the variable highScore is printed, the re-	sult of the calculations is sent back
if (score >= 5000) {	from the calculateScore method and assigned	ot the variable
<pre>do something; } else if (score <1000 && score</pre>		System.ou t.p rin tln ("Your f
>= 500) {	Return variable from method to main:	<pre>inal score is: " + finalS core);</pre>
do something else;	public static int method Nam e(b o	olean dont Paramal sintoria; tPar
] else [am) {	}
do this if all previous conditions are false	}	return -1;
}	Changing $\operatorname{void} to \operatorname{int}$ means we are returnin	gan//i=11tindbitatteyspientcathdevtallaie/not found
	Method	
Method	The method must explicitly return he variable:	Method Overloading
Prevent code duplication	return newValue;	Allows us to create multiple methods with the
Same as 'function' in other languages	An int variable is expected, whether or not the	
Set the name of the method, then the data	(i.e. when an if statement is either true or false	e, something must be returned the method will not signature. The number of paramaters make it
type and name of parameters it will access		5
from the main method		The methods can have the same name but d
Java automatically creates variables with		<pre>sum(int a, int b);</pre>
the appropriate data type for us		<pre>sum(int a, int b, int c); sum(int a, int b, int c, int d);</pre>
Defining a new method:		
public static void method Nam e(b	oolean boolParam int intPar	<pre>public static int sum(int a, int return a + b;</pre>
am) {	oolean boollalam, int intial	}
}		public static int sum(int a, int
void means to "return nothing"		return $a + b + c$;
		}
		public static int sum(int a, int
		return $a + b + c + d;$
		}
		pintln methid is an example of method overlo
		methods of the same name

cheatography.com/bayan-a/

Published 19th July, 2023. Last updated 1st June, 2020. Page 1 of 4.

Measure your website readability!

https://readable.com

Java Mastery - Part 2 Cheat Sheet by Bayan (Bayan.A) via cheatography.com/122738/cs/22936/

Method Overloading (cont)	Switch (cont)		
It improves code readability and re-usability	break;		
Easier to remember one method name	default:		
isntead of many	System.ou t.p rin	tln ("Was not 1 d	or
Achieves naming consistency	2");		
Gives flexibility to call a similar method with	break;		:
different types of data, based on defined	}		1
arguments/parameters	Case tests can also be	on one line. e.a.:	
Switch			
Similar to the if-then-else statement	Common Methods		
	variable.toLowe-	Turns the string	
Good for testing values of the same variable	rCase()	variable to all lower	
The variable that will be changed goes inside the	case		
the conditions tested will be the case numbers	variable.toUppe-	Turns the string	
break; are essential to close off your case compa	k; are essential to close off your case comparing the without it, results with a second s		
The final switch statement is default, same as els	e	UPPERCASE	
Switch statement	if Alath-relue d (qoivalten)	Rounds decimal	
<pre>int value = 1;</pre>	int value = 1;	numbers to the	
switch (value) {	Setting the variable to be	0	
case 1:	String.forma- <u>if</u> (<u>value</u> == 1) { t("%.21" , variable) {	Converts and outputs the variable	
System.ou t.p rin tln ("Value was	System.ou t.p rin		
1");	1");	two decimale points	
break;	Math.sqrt(variable);	Variable must be	
case 2:	} else if (value =	_long. If₁not, add	1
System.ou t.p rin tln ("Value was	System.ou t.p rin	(long) before the tln ("Yalue was	-
2");	2");	expression	
	variable1.equals(v-	Tests if one String	
	ariable2)	variable is equal to	

-		e
ak;		S
		f
		i
e tests can also be o	on one line, e.g.:	
mon Methods		t
e() ets	Turns the string variable to all lower case	ĉ
eWithout it, results	Turns the string Wຢູ່ໄ¦່ຄອເ⊌າຍາຊ¢lictable. UPPERCASE	i
-velue de(quivalden)	Rounds decimal	E
value = 1;	numbers to the	f
g the variable to be g.forma- ^{zalue} == 1) { 2 1 , variable)	nearest value changed Converts and outputs the variable hଧmbelf Waithijestvas	i
	two decimale points	i
se if (value ==	Variable must be long. If not, add (long) before the tln ("Value was expression	L tl
le2)	Tests if one String variable is equal to another	

For Loop/Statement

Processes a condition is	a part of a code block until a s̪at̪isfied
Variable cre System exists in tha	ated in the for statement only 1.ou t.p rin tin ("Was not it code block
Structure:	
for(init; term	nination; increment)
<i>init:</i> case 3	Code initialised once at the case 4:case 5: start of the loop
termin- ation:	Determines at what point it exits the for loop
	Once it evaluates to false, it will exit the loop and proceed to the next line
increment	An expression invoked after each iteration
Example:	
for(int i	i = 0; i < 5; i++)
int i = 0	initialise i to zero,
i < 5	test if i is less than 5 and keeps processing until i is greater than 5
i++	add 1 to the value of i
	wards or backwards depends on

the conditions and ranges

By Bayan (Bayan.A)

cheatography.com/bayan-a/

Published 19th July, 2023. Last updated 1st June, 2020. Page 2 of 4.

Sponsored by Readable.com Measure your website readability! https://readable.com

Java Mastery - Part 2 Cheat Sheet by Bayan (Bayan.A) via cheatography.com/122738/cs/22936/

Can have any number of methods for), while() allows you to loop until the expression is the the count must be incremented, otherwise you finitial loop for version (loop omethod) for (int i = 0; i != 5; i++) { int i = 1 //do something } //do something / //do something } //do something } //do something / //do something / ///do something / //do s	For Statement	Object-C	Priented Programming	Object-Oriented Programming (cont)	Object-Oriented Programming (cont)	
Image: Intervent Interven		Object	A value of a class type	this.model = model; this mean	ns to	
Protected printing transformed is construction of the same behaviour is shown via methods e.g. a dogs name, breed, colour acconstruction of the contents of the parameter acd e.g. a dogs name, breed, colour while/dowhile Behaviour is shown via methods e.g. a dogs name, breed, colour e.g. a dogs name, breed, colour e.g. a dogs name, breed, colour while/dowhile Class Describes set of objects with the same behaviour contents of the parameter acd e.g. a dogs name, breed, colour Similar to the for loop Class Class Describes set of objects with the same behaviour Classes the value of different methods contents of the variables access modifier is set to private for socurity Class variables: Variables delaned within a dass, outside any method, with the static different methods for version while variable saccess modifier is set to private for socurity for version while (int i = 1 reasons, we don't want users to methods class variables: Variables delaned within a dass, outside any method, with the static different method if the variable site private for socurity while (condition { mathing while (int i = 1 reasons, we don't want users to methods don't want watables while (crushing) while (int i = 6) first if the variables i	<pre>for (int number = 1); number < 7); number += 2)</pre>		Has states and behaviours			
a dogs name, breed, colour contents of the parameter model, and the same behaviour: barking, wagging tail, running wagging tail, r			State is stored in fields		el	
image 1			e.g. a dogs name, breed, colour		of the	
Later: 1 e.g. a dogs behaviour: barking, wagging tail, running e.g. private for biocks Class be to	number= 1		Behaviour is shown via methods	paramete	r mod	
Winder downing Others Description Classes and we is the static variables: Variable: Variables: Variable: Variables: Variable: Variables: Variable: Variab					as	
Similar to the for loop Can have any number of methods Can bave any number of methods instead of looping a certain number of times(as seen in critical within) allows you to loop until the expression is true to access the value of different methods Can have any number of methods Cocal variables: Defined in methods, constructors or blocks or false Updating a variable using a method instead Instance variables: Variables declared within a class to utors or blocks Instance variables: Variables declared within a class, outside any method, initialised when the class in instantiated. for version Why? If the variables access modifier is set to private for security reasons, we don't want users to while (i = 0; i != 5; i+i) { int i = 1 is to private for security reasons, we don't want users to while (i = 6 threating) a. before the class to refer to a particular field while condition { //do something}. before the class to refer to a particular field Cha variables while(condition { //do something}. before the class to refer to a particular field Cha variable while(condition { //do something}. before the class to refer to a particular field Cha variable while(crue) { if (i = - 5) { break; } } //do while condition Cha variable Cha variable //dataments semi-colon after //do (Reading user input with Scanner, scanner i declared as a static variable	while/do-while	Class		-		
Updating a variable using a method instead infinite loop Updating a variable using a method instead of directly Instantiated. Class variables. Variables declared within a class outside any method, with the static keyword for version while version infinite loop if the variables access modifier is set to private for security reasons, we don't want users to //do something it 'public' Class variables. Variables declared within a class, outside any method, with the static keyword while int i = 1 int i = 1 set to private for security reasons, we don't want users to //do something it 'public' while int i = 1 particular field Static Variables Declared using the keyword static while(condition { //statements } to count++; aparticular field Static variable mad that yaria will see refressions do {	for), `while() allows you to loop until the		Can have any number of methods to access the value of different	Local variables: Defined in methods, c uctors or blocks		
for version Why? If the variables access modifier is set to private for security class, outside any method, with the static keyword for (int i = 0; i != 5; i++) { int i = 1 reasons, we don't want users to while (i != 6 thirscuty change the variable by { class, outside any method, with the static keyword //do something } while (i != 6 thirscuty change the variable by { making it 'public' Declared using the keyword static //do condition { making it 'public' //do something } Declared using the keyword static //do condition { making it 'public' Mile Declared using the keyword static //do condition { making it 'public' Mile (condition { Mile //do condition { making it 'public' Declared using the keyword static Also known as static member variables //do condition { making it 'public' Mile (condition { Mile Mile //dot condition { making it 'public' Mile (condition) Mile (condition) Mile (condition) //dot { Remember, semi-colon after semi-colon after Mile (condition) Mile condition //dot (count++; making it (count != 6); member variables They don't use the static keyword //do	the count must be incremented, otherwise	you will enter an of direct	g a variable using a method instead y	class in instantiated.		
for (int i = 0; i != 5; i++) { int i = 1 reasons, we don't want users to while (i != 6 tirectly change the variable by { making it 'public' //do something }. before the class to refer to a particular field keyword while particular field Declared using the keyword static while (condition { //statements } } particular field Declared using the keyword static //statements } while (true) { (tru	for version	Why? while version				
//do something } while (i 1 = 6 directly change the variable by { making it 'public' //do something }, before the class to refer to a particular field Static Variables while (condition { ///statements } particular field Declared using the keyword static // Motion { ///statements } making it 'public' Also known as static member variables // Motion { ///statements } making it 'public' Mile (condition { ///statements } // Motion { ///statements } making it 'public' Mile (condition { ///statements } // Motion { ///statements } making it 'public' Mile (condition { ///statements } // Motion { ///statements } making it 'public' Mile (condition { ///statements } // Motion { ///statements } making it 'public' Mile (condition { ///statements } // State as a static variable making user input with Scanner, scanner incelared user a static variable Accessible by static methods directly Reading user input with Scanner, scanner incelared user a static variable e.g. private static String n a not a meri instatic y input with scanner, scanner incelared user a static variable Mile (condition ; // Mule (condition; while condition Instance variables Mile (condition ; // While (condition; while condition Mile (condition ; Mile (condition ;	for(int i = 0; i != 5; i++) {	int i = 1		keyword		
imaking it 'public' //do something it 'public' Declared using the keyword static while particular field Also known as static member variables while(condition { //statements Chart //statements same static variable Every class instance shares the same static variable Chart while(condition {						
while particular field Also known as static member variables while(condition { Every class instance shares the same static variable Chas are static variable //statements while(true) { mad that varia will test to true are static variable mad that varia will test to true are static variable while(true) { fif(1 == 5) { break; count++; Accessible by static methods directly of the instance of a class bo-While thill always execute once or more depending on defined expressions Accessible by static methods directly do { Remember, //statements semi-colon after while condition e.g. private static String na not a me; good class count = 1; do { Instance variables They don't use the static keyword known as fields or member variables Known as fields or member variables		{	making it 'public'			
while particular field Also known as static member variables while(condition { Every class instance shares the same static variable Chas are static variable //statements while(true) { mad that varia will test to true are static variable mad that varia will test to true are static variable while(true) { fif(1 == 5) { break; count++; Accessible by static methods directly of the instance of a class bo-While thill always execute once or more depending on defined expressions Accessible by static methods directly do { Remember, //statements semi-colon after while condition e.g. private static String na not a me; good class count = 1; do { Instance variables They don't use the static keyword known as fields or member variables Known as fields or member variables		//do something	s . before the class to refer to a			
<pre>//statements } while(true){ if(i == 5){ break; } count++; Do-While It will always execute once or more depending on defined expressions do{</pre>	while				S	
Do-While Accessible by static methods directly It will always execute once or more depending on defined expressions Reading user input with Scanner, scanner declared as a static variable do { Remember, //statements semi-colon after while condition while(condition); while condition count = 1; while condition do { Instance variables count ++; They don't use the static keyword while(count != 6); Known as fields or member variables	<pre>while(condition { //statements } while(true) { if(i == 5) { break; } count+t; </pre>				Chang made that variab will be seen i other instan	
It will always execute once or more depending on defined expressions Reading user input with Scanner, scanner declared as a static variable do { Remember, //statements semi-colon after while(condition); while condition count = 1; while condition do { Instance variables count ++; They don't use the static keyword while(count != 6); Known as fields or member variables				Accessible by static methods directly		
do { Remember, e.g. private static String na not a //statements semi-colon after me; good } while(condition); while condition idea count = 1; Instance variables idea do { They don't use the static keyword Known as fields or member variables } while(count != 6); Belong to an instance of a class Belong to an instance of a class	It will always execute once or more deper	nding on defined			nner is	
do { They don't use the static keyword count++; Known as fields or member variables } while(count != 6); Belong to an instance of a class	do { //statements } while(condition);	semi-colon after	r	• •	not a good idea	
do {They don't use the static keywordcount++;Known as fields or member variables} while (count != 6);Belong to an instance of a class	count = 1;			Instance variables		
count++; Known as fields or member variables Belong to an instance of a class	do {					
Belong to an instance of a class						
	<pre>} while(count != 6);</pre>					
				Represents the state of an isntance		

By Bayan (Bayan.A) cheatography.com/bayan-a/ Published 19th July, 2023. Last updated 1st June, 2020. Page 3 of 4. Sponsored by Readable.com

https://readable.com

Measure your website readability!

Java Mastery - Part 2 Cheat Sheet by Bayan (Bayan.A) via cheatography.com/122738/cs/22936/

Instance va	riables (cont)
Every instance:	-Has it's own copy of an instance variable -Can have a different value(- state)
Static Meth	od
Declared u	sing a static modifier
	ss instance methods and riables directly
Cannot use the this keyword	Usually used for operations that don't require data from an instance of the class (from thi s)
	thod does not use instance hat method should be declared nethod
0	a static method and is called by it starts an application
Example	
a, int 1	ou t.p rin tln ("sum = "
Instance M	ethods
Belongs to	an instance of a class
-	must instantiate the class first

To use, we must instantiate the class first by using the new keyword

Can access instance methods and variables directly

Can access static methods and static variables directly

Example:

```
class Dog {
  public void bark() {
   System.ou t.p rin tln ("wo of")
;
```

} }

By Bayan (Bayan.A) cheatography.com/bayan-a/ Published 19th July, 2023. Last updated 1st June, 2020. Page 4 of 4.

Sponsored by Readable.com Measure your website readability! https://readable.com

public static void main(S tring [] args) { Dog rex = new Dog(); //create insta nce `rex.b ark(); //call instance method` } } Static or Instance Method? Static or Instance Method? Static or Instance Method? (south static) (south

Should a method be static?

Instance Methods (cont)
public class Main {

```
|
V
```

Doies it use any fields(instance variables) or Instance methods?

YES? It should probably be an instance method

NO? It should probably be a static method