

# **Washington State University**



## **422 Deliverable 2**

### **Test Cases**

#### **Team members**

Yuer Shen (SID: 11421959)

Courtney Snyder (SID: 11463078)

Ran Tao (SID: 11488080)

**CptS 422 Software Engineering Principle II**

**Spring 2017**

**Instructor: Venera Arnaoudova**

**Black Box Test:** Test cases for operation and operator only

### Equivalence Class Partitioning Testing

Operand:

- Valid:
  - Object: string, user defined object, defined in a JAR, etc.
  - Alphanumeric: char, etc.
  - Numeric: int, double, float, short, long, and etc.
- Invalid: Otherwise.

Operator:

- Valid: +, -, \*, /, %, ++, --, ==, !=, >, <, >=, <=, &, |, ^, ~, <<, >>, >>>, &&, ||, !, =, +=, -=, \*=, /=, %=, <<=, >>=, &=, ^=, |=.
- Invalid: Otherwise (===, --- and etc).

For sake of generality, we will use Object class as our object.

### SECT Table

#### Case 1: single operand

Total cases: operand \* operator = 4 \* 35 = 140

#### Test Frames:

	operand	operator
SECT1	object	+
SECT2	object	-
SECT3	object	*
SECT4	object	/
SECT5	object	%
SECT6	object	++
SECT7	object	--
SECT8	object	==
SECT9	object	!=
SECT10	object	>

SECT11	object	<
SECT12	object	>=
SECT13	object	<=
SECT14	object	&
SECT15	object	
SECT16	object	^
SECT17	object	~
SECT18	object	<<
SECT19	object	>>
SECT20	object	>>>
SECT21	object	&&
SECT22	object	
SECT23	object	!
SECT24	object	=
SECT25	object	+=
SECT26	object	-=
SECT27	object	*=
SECT28	object	/=
SECT29	object	%=
SECT30	object	<<=
SECT31	object	>>=
SECT32	object	&=
SECT33	object	^=
SECT34	object	=
SECT35	alpha	+
SECT36	alpha	-

SECT37	alpha	*
SECT38	alpha	/
SECT39	alpha	%
SECT40	alpha	++
SECT41	alpha	--
SECT42	alpha	==
SECT43	alpha	!=
SECT44	alpha	>
SECT45	alpha	<
SECT46	alpha	>=
SECT47	alpha	<=
SECT48	alpha	&
SECT49	alpha	
SECT50	alpha	^
SECT51	alpha	~
SECT52	alpha	<<
SECT53	alpha	>>
SECT54	alpha	>>>
SECT55	alpha	&&
SECT56	alpha	
SECT57	alpha	!
SECT58	alpha	=
SECT59	alpha	+=
SECT60	alpha	-=
SECT61	alpha	*=
SECT62	alpha	/=

SECT63	alpha	%=
SECT64	alpha	<<=
SECT65	alpha	>>=
SECT66	alpha	&=
SECT67	alpha	^=
SECT68	alpha	=
SECT69	number	+
SECT70	number	-
SECT71	number	*
SECT72	number	/
SECT73	number	%
SECT74	number	++
SECT75	number	--
SECT76	number	==
SECT77	number	!=
SECT78	number	>
SECT79	number	<
SECT80	number	>=
SECT81	number	<=
SECT82	number	&
SECT83	number	
SECT84	number	^
SECT85	number	~
SECT86	number	<<
SECT87	number	>>
SECT88	number	>>>

SECT89	number	&&
SECT90	number	
SECT91	number	!
SECT92	number	=
SECT93	number	+=
SECT94	number	-=
SECT95	number	*=
SECT96	number	/=
SECT97	number	%=
SECT98	number	<<=
SECT99	number	>>=
SECT100	number	&=
SECT101	number	^=
SECT102	number	=
SECT103	Otherwise	+
SECT104	Otherwise	-
SECT105	Otherwise	*
SECT106	Otherwise	/
SECT107	Otherwise	%
SECT108	Otherwise	++
SECT109	Otherwise	--
SECT110	Otherwise	==
SECT111	Otherwise	!=
SECT112	Otherwise	>
SECT113	Otherwise	<
SECT114	Otherwise	>=

SECT115	Otherwise	<=
SECT116	Otherwise	&
SECT117	Otherwise	
SECT118	Otherwise	^
SECT119	Otherwise	~
SECT120	Otherwise	<<
SECT121	Otherwise	>>
SECT122	Otherwise	>>>
SECT123	Otherwise	&&
SECT124	Otherwise	
SECT125	Otherwise	!
SECT126	Otherwise	=
SECT127	Otherwise	+=
SECT128	Otherwise	-=
SECT129	Otherwise	*=
SECT130	Otherwise	/=
SECT131	Otherwise	%=
SECT132	Otherwise	<<=
SECT133	Otherwise	>>=
SECT134	Otherwise	&=
SECT135	Otherwise	^=
SECT136	Otherwise	=
SECT137	object	Otherwise
SECT138	alpha	Otherwise
SECT139	number	Otherwise
SECT140	Otherwise	Otherwise

**Test Cases:**

	Operand	Operator	VALID/INVALID
SECT1	"Hello"	+	INVALID
SECT2	"Hello"	-	INVALID
SECT3	"Hello"	*	INVALID
SECT4	"Hello"	/	INVALID
SECT5	"Hello"	%	INVALID
SECT6	"Hello"	++	VALID
SECT7	"Hello"	--	VALID
SECT8	"Hello"	==	INVALID
SECT9	"Hello"	!=	INVALID
SECT10	"Hello"	>	INVALID
SECT11	"Hello"	<	INVALID
SECT12	"Hello"	>=	INVALID
SECT13	"Hello"	<=	INVALID
SECT14	"Hello"	&	INVALID
SECT15	"Hello"		INVALID
SECT16	"Hello"	^	INVALID



SECT17	"Hello"	~	VALID
SECT18	"Hello"	<<	INVALID
SECT19	"Hello"	>>	INVALID
SECT20	"Hello"	>>>	INVALID
SECT21	"Hello"	&&	INVALID
SECT22	"Hello"		INVALID
SECT23	"Hello"	!	VALID
SECT24	"Hello"	=	INVALID
SECT25	"Hello"	+=	INVALID
SECT26	"Hello"	-=	INVALID
SECT27	"Hello"	*=	INVALID
SECT28	"Hello"	/=	INVALID
SECT29	"Hello"	%=	INVALID
SECT30	"Hello"	<<=	INVALID
SECT31	"Hello"	>>=	INVALID
SECT32	"Hello"	&=	INVALID
SECT33	"Hello"	^=	INVALID
SECT34	"Hello"	=	INVALID

SECT35	'c'	+	INVALID
SECT36	'c'	-	INVALID
SECT37	'c'	*	INVALID
SECT38	'c'	/	INVALID
SECT39	'c'	%	INVALID
SECT40	'c'	++	VALID
SECT41	'c'	--	VALID
SECT42	'c'	==	INVALID
SECT43	'c'	!=	INVALID
SECT44	'c'	>	INVALID
SECT45	'c'	<	INVALID
SECT46	'c'	>=	INVALID
SECT47	'c'	<=	INVALID
SECT48	'c'	&	INVALID
SECT49	'c'		INVALID
SECT50	'c'	^	INVALID
SECT51	'c'	~	VALID
SECT52	'c'	<<	INVALID

SECT53	'c'	>>	INVALID
SECT54	'c'	>>>	INVALID
SECT55	'c'	&&	INVALID
SECT56	'c'		INVALID
SECT57	'c'	!	VALID
SECT58	'c'	=	INVALID
SECT59	'c'	+=	INVALID
SECT60	'c'	-=	INVALID
SECT61	'c'	*=	INVALID
SECT62	'c'	/=	INVALID
SECT63	'c'	%=	INVALID
SECT64	'c'	<<=	INVALID
SECT65	'c'	>>=	INVALID
SECT66	'c'	&=	INVALID
SECT67	'c'	^=	INVALID
SECT68	'c'	=	INVALID
SECT69	42	+	INVALID
SECT70	42	-	INVALID

SECT71	42	*	INVALID
SECT72	42	/	INVALID
SECT73	42	%	INVALID
SECT74	42	++	VALID
SECT75	42	--	VALID
SECT76	42	==	INVALID
SECT77	42	!=	INVALID
SECT78	42	>	INVALID
SECT79	42	<	INVALID
SECT80	42	>=	INVALID
SECT81	42	<=	INVALID
SECT82	42	&	INVALID
SECT83	42		INVALID
SECT84	42	^	INVALID
SECT85	42	~	VALID
SECT86	42	<<	INVALID
SECT87	42	>>	INVALID
SECT88	42	>>>	INVALID

SECT89	42	&&	INVALID
SECT90	42		INVALID
SECT91	42	!	VALID
SECT92	42	=	INVALID
SECT93	42	+=	INVALID
SECT94	42	-=	INVALID
SECT95	42	*=	INVALID
SECT96	42	/=	INVALID
SECT97	42	%=	INVALID
SECT98	42	<<=	INVALID
SECT99	42	>>=	INVALID
SECT100	42	&=	INVALID
SECT101	42	^=	INVALID
SECT102	42	=	INVALID
SECT103	中文	+	INVALID
SECT104	中文	-	INVALID
SECT105	中文	*	INVALID
SECT106	中文	/	INVALID

SECT107	中文	%	INVALID
SECT108	中文	++	INVALID
SECT109	中文	--	INVALID
SECT110	中文	==	INVALID
SECT111	中文	!=	INVALID
SECT112	中文	>	INVALID
SECT113	中文	<	INVALID
SECT114	中文	>=	INVALID
SECT115	中文	<=	INVALID
SECT116	中文	&	INVALID
SECT117	中文		INVALID
SECT118	中文	^	INVALID
SECT119	中文	~	INVALID
SECT120	中文	<<	INVALID
SECT121	中文	>>	INVALID
SECT122	中文	>>>	INVALID
SECT123	中文	&&	INVALID
SECT124	中文		INVALID

SECT125	中文	!	INVALID
SECT126	中文	=	INVALID
SECT127	中文	+=	INVALID
SECT128	中文	-=	INVALID
SECT129	中文	*=	INVALID
SECT130	中文	/=	INVALID
SECT131	中文	%=	INVALID
SECT132	中文	<<=	INVALID
SECT133	中文	>>=	INVALID
SECT134	中文	&=	INVALID
SECT135	中文	^=	INVALID
SECT136	中文	=	INVALID
SECT137	"Hello"	===	INVALID
SECT138	'c'	===	INVALID
SECT139	42	===	INVALID
SECT140	中文	===	INVALID

### Case 2: double operand

Total cases: operand \* operand \* operator = 4 \* 4 \* 35 = 560

**Test Frames:**

	Operand	Operator	Operand
SECT1	object	+	object
SECT2	object	-	object
SECT3	object	*	object
SECT4	object	/	object
SECT5	object	%	object
SECT6	object	++	object
SECT7	object	--	object
SECT8	object	==	object
SECT9	object	!=	object
SECT10	object	>	object
SECT11	object	<	object
SECT12	object	>=	object
SECT13	object	<=	object
SECT14	object	&	object
SECT15	object		object
SECT16	object	^	object



SECT17	object	~	object
SECT18	object	<<	object
SECT19	object	>>	object
SECT20	object	>>>	object
SECT21	object	&&	object
SECT22	object		object
SECT23	object	!	object
SECT24	object	=	object
SECT25	object	+=	object
SECT26	object	-=	object
SECT27	object	*=	object
SECT28	object	/=	object
SECT29	object	%=	object
SECT30	object	<<=	object
SECT31	object	>>=	object
SECT32	object	&=	object
SECT33	object	^=	object
SECT34	object	=	object

SECT35	alpha	+	object
SECT36	alpha	-	object
SECT37	alpha	*	object
SECT38	alpha	/	object
SECT39	alpha	%	object
SECT40	alpha	++	object
SECT41	alpha	--	object
SECT42	alpha	==	object
SECT43	alpha	!=	object
SECT44	alpha	>	object
SECT45	alpha	<	object
SECT46	alpha	>=	object
SECT47	alpha	<=	object
SECT48	alpha	&	object
SECT49	alpha		object
SECT50	alpha	^	object
SECT51	alpha	~	object
SECT52	alpha	<<	object

SECT53	alpha	>>	object
SECT54	alpha	>>>	object
SECT55	alpha	&&	object
SECT56	alpha		object
SECT57	alpha	!	object
SECT58	alpha	=	object
SECT59	alpha	+=	object
SECT60	alpha	-=	object
SECT61	alpha	*=	object
SECT62	alpha	/=	object
SECT63	alpha	%=	object
SECT64	alpha	<<=	object
SECT65	alpha	>>=	object
SECT66	alpha	&=	object
SECT67	alpha	^=	object
SECT68	alpha	=	object
SECT69	number	+	object
SECT70	number	-	object

SECT71	number	*	object
SECT72	number	/	object
SECT73	number	%	object
SECT74	number	++	object
SECT75	number	--	object
SECT76	number	==	object
SECT77	number	!=	object
SECT78	number	>	object
SECT79	number	<	object
SECT80	number	>=	object
SECT81	number	<=	object
SECT82	number	&	object
SECT83	number		object
SECT84	number	^	object
SECT85	number	~	object
SECT86	number	<<	object
SECT87	number	>>	object
SECT88	number	>>>	object

SECT89	number	&&	object
SECT90	number		object
SECT91	number	!	object
SECT92	number	=	object
SECT93	number	+=	object
SECT94	number	-=	object
SECT95	number	*=	object
SECT96	number	/=	object
SECT97	number	%=	object
SECT98	number	<<=	object
SECT99	number	>>=	object
SECT100	number	&=	object
SECT101	number	^=	object
SECT102	number	=	object
SECT103	Otherwise	+	object
SECT104	Otherwise	-	object
SECT105	Otherwise	*	object
SECT106	Otherwise	/	object

SECT107	Otherwise	%	object
SECT108	Otherwise	++	object
SECT109	Otherwise	--	object
SECT110	Otherwise	==	object
SECT111	Otherwise	!=	object
SECT112	Otherwise	>	object
SECT113	Otherwise	<	object
SECT114	Otherwise	>=	object
SECT115	Otherwise	<=	object
SECT116	Otherwise	&	object
SECT117	Otherwise		object
SECT118	Otherwise	^	object
SECT119	Otherwise	~	object
SECT120	Otherwise	<<	object
SECT121	Otherwise	>>	object
SECT122	Otherwise	>>>	object
SECT123	Otherwise	&&	object
SECT124	Otherwise		object

SECT125	Otherwise	!	object
SECT126	Otherwise	=	object
SECT127	Otherwise	+=	object
SECT128	Otherwise	-=	object
SECT129	Otherwise	*=	object
SECT130	Otherwise	/=	object
SECT131	Otherwise	%=	object
SECT132	Otherwise	<<=	object
SECT133	Otherwise	>>=	object
SECT134	Otherwise	&=	object
SECT135	Otherwise	^=	object
SECT136	Otherwise	=	object
SECT137	object	Otherwise	object
SECT138	alpha	Otherwise	object
SECT139	number	Otherwise	object
SECT140	Otherwise	Otherwise	object
SECT141	object	+	alpha
SECT142	object	-	alpha

SECT143	object	*	alpha
SECT144	object	/	alpha
SECT145	object	%	alpha
SECT146	object	++	alpha
SECT147	object	--	alpha
SECT148	object	==	alpha
SECT149	object	!=	alpha
SECT150	object	>	alpha
SECT151	object	<	alpha
SECT152	object	>=	alpha
SECT153	object	<=	alpha
SECT154	object	&	alpha
SECT155	object		alpha
SECT156	object	^	alpha
SECT157	object	~	alpha
SECT158	object	<<	alpha
SECT159	object	>>	alpha
SECT160	object	>>>	alpha



SECT161	object	&&	alpha
SECT162	object		alpha
SECT163	object	!	alpha
SECT164	object	=	alpha
SECT165	object	+=	alpha
SECT166	object	-=	alpha
SECT167	object	*=	alpha
SECT168	object	/=	alpha
SECT169	object	%=	alpha
SECT170	object	<<=	alpha
SECT171	object	>>=	alpha
SECT172	object	&=	alpha
SECT173	object	^=	alpha
SECT174	object	=	alpha
SECT175	alpha	+	alpha
SECT176	alpha	-	alpha
SECT177	alpha	*	alpha
SECT178	alpha	/	alpha

SECT179	alpha	%	alpha
SECT180	alpha	++	alpha
SECT181	alpha	--	alpha
SECT182	alpha	==	alpha
SECT183	alpha	!=	alpha
SECT184	alpha	>	alpha
SECT185	alpha	<	alpha
SECT186	alpha	>=	alpha
SECT187	alpha	<=	alpha
SECT188	alpha	&	alpha
SECT189	alpha		alpha
SECT190	alpha	^	alpha
SECT191	alpha	~	alpha
SECT192	alpha	<<	alpha
SECT193	alpha	>>	alpha
SECT194	alpha	>>>	alpha
SECT195	alpha	&&	alpha
SECT196	alpha		alpha

SECT197	alpha	!	alpha
SECT198	alpha	=	alpha
SECT199	alpha	+=	alpha
SECT200	alpha	-=	alpha
SECT201	alpha	*=	alpha
SECT202	alpha	/=	alpha
SECT203	alpha	%=	alpha
SECT204	alpha	<<=	alpha
SECT205	alpha	>>=	alpha
SECT206	alpha	&=	alpha
SECT207	alpha	^=	alpha
SECT208	alpha	=	alpha
SECT209	number	+	alpha
SECT210	number	-	alpha
SECT211	number	*	alpha
SECT212	number	/	alpha
SECT213	number	%	alpha
SECT214	number	++	alpha

SECT215	number	--	alpha
SECT216	number	==	alpha
SECT217	number	!=	alpha
SECT218	number	>	alpha
SECT219	number	<	alpha
SECT220	number	>=	alpha
SECT221	number	<=	alpha
SECT222	number	&	alpha
SECT223	number		alpha
SECT224	number	^	alpha
SECT225	number	~	alpha
SECT226	number	<<	alpha
SECT227	number	>>	alpha
SECT228	number	>>>	alpha
SECT229	number	&&	alpha
SECT230	number		alpha
SECT231	number	!	alpha
SECT232	number	=	alpha

SECT233	number	$+=$	alpha
SECT234	number	$-=$	alpha
SECT235	number	$*=$	alpha
SECT236	number	$/=$	alpha
SECT237	number	$\%=$	alpha
SECT238	number	$<<=$	alpha
SECT239	number	$>>=$	alpha
SECT240	number	$\&=$	alpha
SECT241	number	$\wedge=$	alpha
SECT242	number	$ =$	alpha
SECT243	Otherwise	$+$	alpha
SECT244	Otherwise	$-$	alpha
SECT245	Otherwise	$*$	alpha
SECT246	Otherwise	$/$	alpha
SECT247	Otherwise	$\%$	alpha
SECT248	Otherwise	$++$	alpha
SECT249	Otherwise	$--$	alpha
SECT250	Otherwise	$==$	alpha

SECT251	Otherwise	!=	alpha
SECT252	Otherwise	>	alpha
SECT253	Otherwise	<	alpha
SECT254	Otherwise	>=	alpha
SECT255	Otherwise	<=	alpha
SECT256	Otherwise	&	alpha
SECT257	Otherwise		alpha
SECT258	Otherwise	^	alpha
SECT259	Otherwise	~	alpha
SECT260	Otherwise	<<	alpha
SECT261	Otherwise	>>	alpha
SECT262	Otherwise	>>>	alpha
SECT263	Otherwise	&&	alpha
SECT264	Otherwise		alpha
SECT265	Otherwise	!	alpha
SECT266	Otherwise	=	alpha
SECT267	Otherwise	+=	alpha
SECT268	Otherwise	-=	alpha

SECT269	Otherwise	*=	alpha
SECT270	Otherwise	/=	alpha
SECT271	Otherwise	%=	alpha
SECT272	Otherwise	<<=	alpha
SECT273	Otherwise	>>=	alpha
SECT274	Otherwise	&=	alpha
SECT275	Otherwise	^=	alpha
SECT276	Otherwise	=	alpha
SECT277	object	Otherwise	alpha
SECT278	alpha	Otherwise	alpha
SECT279	number	Otherwise	alpha
SECT280	Otherwise	Otherwise	alpha
SECT281	object	+	number
SECT282	object	-	number
SECT283	object	*	number
SECT284	object	/	number
SECT285	object	%	number
SECT286	object	++	number

SECT287	object	--	number
SECT288	object	==	number
SECT289	object	!=	number
SECT290	object	>	number
SECT291	object	<	number
SECT292	object	>=	number
SECT293	object	<=	number
SECT294	object	&	number
SECT295	object		number
SECT296	object	^	number
SECT297	object	~	number
SECT298	object	<<	number
SECT299	object	>>	number
SECT300	object	>>>	number
SECT301	object	&&	number
SECT302	object		number
SECT303	object	!	number
SECT304	object	=	number



SECT305	object	+=	number
SECT306	object	-=	number
SECT307	object	*=	number
SECT308	object	/=	number
SECT309	object	%=	number
SECT310	object	<<=	number
SECT311	object	>>=	number
SECT312	object	&=	number
SECT313	object	^=	number
SECT314	object	=	number
SECT315	alpha	+	number
SECT316	alpha	-	number
SECT317	alpha	*	number
SECT318	alpha	/	number
SECT319	alpha	%	number
SECT320	alpha	++	number
SECT321	alpha	--	number
SECT322	alpha	==	number

SECT323	alpha	!=	number
SECT324	alpha	>	number
SECT325	alpha	<	number
SECT326	alpha	>=	number
SECT327	alpha	<=	number
SECT328	alpha	&	number
SECT329	alpha		number
SECT330	alpha	^	number
SECT331	alpha	~	number
SECT332	alpha	<<	number
SECT333	alpha	>>	number
SECT334	alpha	>>>	number
SECT335	alpha	&&	number
SECT336	alpha		number
SECT337	alpha	!	number
SECT338	alpha	=	number
SECT339	alpha	+=	number
SECT340	alpha	-=	number

SECT341	alpha	*=	number
SECT342	alpha	/=	number
SECT343	alpha	%=	number
SECT344	alpha	<<=	number
SECT345	alpha	>>=	number
SECT346	alpha	&=	number
SECT347	alpha	^=	number
SECT348	alpha	=	number
SECT349	number	+	number
SECT350	number	-	number
SECT351	number	*	number
SECT352	number	/	number
SECT353	number	%	number
SECT354	number	++	number
SECT355	number	--	number
SECT356	number	==	number
SECT357	number	!=	number
SECT358	number	>	number

SECT359	number	<	number
SECT360	number	>=	number
SECT361	number	<=	number
SECT362	number	&	number
SECT363	number		number
SECT364	number	^	number
SECT365	number	~	number
SECT366	number	<<	number
SECT367	number	>>	number
SECT368	number	>>>	number
SECT369	number	&&	number
SECT370	number		number
SECT371	number	!	number
SECT372	number	=	number
SECT373	number	+=	number
SECT374	number	-=	number
SECT375	number	*=	number
SECT376	number	/=	number

SECT377	number	%=	number
SECT378	number	<<=	number
SECT379	number	>>=	number
SECT380	number	&=	number
SECT381	number	^=	number
SECT382	number	=	number
SECT383	Otherwise	+	number
SECT384	Otherwise	-	number
SECT385	Otherwise	*	number
SECT386	Otherwise	/	number
SECT387	Otherwise	%	number
SECT388	Otherwise	++	number
SECT389	Otherwise	--	number
SECT390	Otherwise	==	number
SECT391	Otherwise	!=	number
SECT392	Otherwise	>	number
SECT393	Otherwise	<	number
SECT394	Otherwise	>=	number

SECT395	Otherwise	<=	number
SECT396	Otherwise	&	number
SECT397	Otherwise		number
SECT398	Otherwise	^	number
SECT399	Otherwise	~	number
SECT400	Otherwise	<<	number
SECT401	Otherwise	>>	number
SECT402	Otherwise	>>>	number
SECT403	Otherwise	&&	number
SECT404	Otherwise		number
SECT405	Otherwise	!	number
SECT406	Otherwise	=	number
SECT407	Otherwise	+=	number
SECT408	Otherwise	-=	number
SECT409	Otherwise	*=	number
SECT410	Otherwise	/=	number
SECT411	Otherwise	%=	number
SECT412	Otherwise	<<=	number

SECT413	Otherwise	>>=	number
SECT414	Otherwise	&=	number
SECT415	Otherwise	^=	number
SECT416	Otherwise	=	number
SECT417	object	Otherwise	number
SECT418	alpha	Otherwise	number
SECT419	number	Otherwise	number
SECT420	Otherwise	Otherwise	number
SECT421	object	+	otherwise
SECT422	object	-	otherwise
SECT423	object	*	otherwise
SECT424	object	/	otherwise
SECT425	object	%	otherwise
SECT426	object	++	otherwise
SECT427	object	--	otherwise
SECT428	object	==	otherwise
SECT429	object	!=	otherwise
SECT430	object	>	otherwise

SECT431	object	<	otherwise
SECT432	object	>=	otherwise
SECT433	object	<=	otherwise
SECT434	object	&	otherwise
SECT435	object		otherwise
SECT436	object	^	otherwise
SECT437	object	~	otherwise
SECT438	object	<<	otherwise
SECT439	object	>>	otherwise
SECT440	object	>>>	otherwise
SECT441	object	&&	otherwise
SECT442	object		otherwise
SECT443	object	!	otherwise
SECT444	object	=	otherwise
SECT445	object	+=	otherwise
SECT446	object	-=	otherwise
SECT447	object	*=	otherwise
SECT448	object	/=	otherwise



SECT449	object	%=	otherwise
SECT450	object	<<=	otherwise
SECT451	object	>>=	otherwise
SECT452	object	&=	otherwise
SECT453	object	^=	otherwise
SECT454	object	=	otherwise
SECT455	alpha	+	otherwise
SECT456	alpha	-	otherwise
SECT457	alpha	*	otherwise
SECT458	alpha	/	otherwise
SECT459	alpha	%	otherwise
SECT460	alpha	++	otherwise
SECT461	alpha	--	otherwise
SECT462	alpha	==	otherwise
SECT463	alpha	!=	otherwise
SECT464	alpha	>	otherwise
SECT465	alpha	<	otherwise
SECT466	alpha	>=	otherwise

SECT467	alpha	<=	otherwise
SECT468	alpha	&	otherwise
SECT469	alpha		otherwise
SECT470	alpha	^	otherwise
SECT471	alpha	~	otherwise
SECT472	alpha	<<	otherwise
SECT473	alpha	>>	otherwise
SECT474	alpha	>>>	otherwise
SECT475	alpha	&&	otherwise
SECT476	alpha		otherwise
SECT477	alpha	!	otherwise
SECT478	alpha	=	otherwise
SECT479	alpha	+=	otherwise
SECT480	alpha	-=	otherwise
SECT481	alpha	*=	otherwise
SECT482	alpha	/=	otherwise
SECT483	alpha	%=	otherwise
SECT484	alpha	<<=	otherwise

SECT485	alpha	>>=	otherwise
SECT486	alpha	&=	otherwise
SECT487	alpha	^=	otherwise
SECT488	alpha	=	otherwise
SECT489	number	+	otherwise
SECT490	number	-	otherwise
SECT491	number	*	otherwise
SECT492	number	/	otherwise
SECT493	number	%	otherwise
SECT494	number	++	otherwise
SECT495	number	--	otherwise
SECT496	number	==	otherwise
SECT497	number	!=	otherwise
SECT498	number	>	otherwise
SECT499	number	<	otherwise
SECT500	number	>=	otherwise
SECT501	number	<=	otherwise
SECT502	number	&	otherwise

SECT503	number		otherwise
SECT504	number	^	otherwise
SECT505	number	~	otherwise
SECT506	number	<<	otherwise
SECT507	number	>>	otherwise
SECT508	number	>>>	otherwise
SECT509	number	&&	otherwise
SECT510	number		otherwise
SECT511	number	!	otherwise
SECT512	number	=	otherwise
SECT513	number	+=	otherwise
SECT514	number	-=	otherwise
SECT515	number	*=	otherwise
SECT516	number	/=	otherwise
SECT517	number	%=	otherwise
SECT518	number	<<=	otherwise
SECT519	number	>>=	otherwise
SECT520	number	&=	otherwise

SECT521	number	$\wedge=$	otherwise
SECT522	number	$ =$	otherwise
SECT523	Otherwise	+	otherwise
SECT524	Otherwise	-	otherwise
SECT525	Otherwise	*	otherwise
SECT526	Otherwise	/	otherwise
SECT527	Otherwise	%	otherwise
SECT528	Otherwise	++	otherwise
SECT529	Otherwise	--	otherwise
SECT530	Otherwise	==	otherwise
SECT531	Otherwise	!=	otherwise
SECT532	Otherwise	>	otherwise
SECT533	Otherwise	<	otherwise
SECT534	Otherwise	>=	otherwise
SECT535	Otherwise	<=	otherwise
SECT536	Otherwise	&	otherwise
SECT537	Otherwise		otherwise
SECT538	Otherwise	$\wedge$	otherwise

SECT539	Otherwise	~	otherwise
SECT540	Otherwise	<<	otherwise
SECT541	Otherwise	>>	otherwise
SECT542	Otherwise	>>>	otherwise
SECT543	Otherwise	&&	otherwise
SECT544	Otherwise		otherwise
SECT545	Otherwise	!	otherwise
SECT546	Otherwise	=	otherwise
SECT547	Otherwise	+=	otherwise
SECT548	Otherwise	-=	otherwise
SECT549	Otherwise	*=	otherwise
SECT550	Otherwise	/=	otherwise
SECT551	Otherwise	%=	otherwise
SECT552	Otherwise	<<=	otherwise
SECT553	Otherwise	>>=	otherwise
SECT554	Otherwise	&=	otherwise
SECT555	Otherwise	^=	otherwise
SECT556	Otherwise	=	otherwise

SECT557	object	Otherwise	otherwise
SECT558	alpha	Otherwise	otherwise
SECT559	number	Otherwise	otherwise
SECT560	Otherwise	Otherwise	otherwise

### Test Cases:

	Operand	Operator	Operand	VALID/INVALID
SECT1	"Hello"	+	"World"	VALID
SECT2	"Hello"	-	"World"	VALID
SECT3	"Hello"	*	"World"	VALID
SECT4	"Hello"	/	"World"	VALID
SECT5	"Hello"	%	"World"	VALID
SECT6	"Hello"	++	"World"	VALID
SECT7	"Hello"	--	"World"	VALID
SECT8	"Hello"	==	"World"	VALID
SECT9	"Hello"	!=	"World"	VALID
SECT10	"Hello"	>	"World"	VALID
SECT11	"Hello"	<	"World"	VALID
SECT12	"Hello"	>=	"World"	VALID

SECT13	"Hello"	<=	"World"	VALID
SECT14	"Hello"	&	"World"	VALID
SECT15	"Hello"		"World"	VALID
SECT16	"Hello"	^	"World"	VALID
SECT17	"Hello"	~	"World"	VALID
SECT18	"Hello"	<<	"World"	VALID
SECT19	"Hello"	>>	"World"	VALID
SECT20	"Hello"	>>>	"World"	VALID
SECT21	"Hello"	&&	"World"	VALID
SECT22	"Hello"		"World"	VALID
SECT23	"Hello"	!	"World"	VALID
SECT24	"Hello"	=	"World"	VALID
SECT25	"Hello"	+=	"World"	VALID
SECT26	"Hello"	-=	"World"	VALID
SECT27	"Hello"	*=	"World"	VALID
SECT28	"Hello"	/=	"World"	VALID
SECT29	"Hello"	%=	"World"	VALID
SECT30	"Hello"	<<=	"World"	VALID



SECT31	"Hello"	>>=	"World"	VALID
SECT32	"Hello"	&=	"World"	VALID
SECT33	"Hello"	^=	"World"	VALID
SECT34	"Hello"	=	"World"	VALID
SECT35	'c'	+	"World"	VALID
SECT36	'c'	-	"World"	VALID
SECT37	'c'	*	"World"	VALID
SECT38	'c'	/	"World"	VALID
SECT39	'c'	%	"World"	VALID
SECT40	'c'	++	"World"	VALID
SECT41	'c'	--	"World"	VALID
SECT42	'c'	==	"World"	VALID
SECT43	'c'	!=	"World"	VALID
SECT44	'c'	>	"World"	VALID
SECT45	'c'	<	"World"	VALID
SECT46	'c'	>=	"World"	VALID
SECT47	'c'	<=	"World"	VALID
SECT48	'c'	&	"World"	VALID

SECT49	'c'		"World"	VALID
SECT50	'c'	^	"World"	VALID
SECT51	'c'	~	"World"	VALID
SECT52	'c'	<<	"World"	VALID
SECT53	'c'	>>	"World"	VALID
SECT54	'c'	>>>	"World"	VALID
SECT55	'c'	&&	"World"	VALID
SECT56	'c'		"World"	VALID
SECT57	'c'	!	"World"	VALID
SECT58	'c'	=	"World"	VALID
SECT59	'c'	+=	"World"	VALID
SECT60	'c'	-=	"World"	VALID
SECT61	'c'	*=	"World"	VALID
SECT62	'c'	/=	"World"	VALID
SECT63	'c'	%=	"World"	VALID
SECT64	'c'	<<=	"World"	VALID
SECT65	'c'	>>=	"World"	VALID
SECT66	'c'	&=	"World"	VALID

SECT67	'c'	$\wedge=$	"World"	VALID
SECT68	'c'	$ =$	"World"	VALID
SECT69	42	+	"World"	VALID
SECT70	42	-	"World"	VALID
SECT71	42	*	"World"	VALID
SECT72	42	/	"World"	VALID
SECT73	42	%	"World"	VALID
SECT74	42	++	"World"	VALID
SECT75	42	--	"World"	VALID
SECT76	42	==	"World"	VALID
SECT77	42	!=	"World"	VALID
SECT78	42	>	"World"	VALID
SECT79	42	<	"World"	VALID
SECT80	42	>=	"World"	VALID
SECT81	42	<=	"World"	VALID
SECT82	42	&	"World"	VALID
SECT83	42		"World"	VALID
SECT84	42	$\wedge$	"World"	VALID

SECT85	42	~	"World"	VALID
SECT86	42	<<	"World"	VALID
SECT87	42	>>	"World"	VALID
SECT88	42	>>>	"World"	VALID
SECT89	42	&&	"World"	VALID
SECT90	42		"World"	VALID
SECT91	42	!	"World"	VALID
SECT92	42	=	"World"	VALID
SECT93	42	+=	"World"	VALID
SECT94	42	-=	"World"	VALID
SECT95	42	*=	"World"	VALID
SECT96	42	/=	"World"	VALID
SECT97	42	%=	"World"	VALID
SECT98	42	<<=	"World"	VALID
SECT99	42	>>=	"World"	VALID
SECT100	42	&=	"World"	VALID
SECT101	42	^=	"World"	VALID
SECT102	42	=	"World"	VALID

SECT103	中文	+	"World"	INVALID
SECT104	中文	-	"World"	INVALID
SECT105	中文	*	"World"	INVALID
SECT106	中文	/	"World"	INVALID
SECT107	中文	%	"World"	INVALID
SECT108	中文	++	"World"	INVALID
SECT109	中文	--	"World"	INVALID
SECT110	中文	==	"World"	INVALID
SECT111	中文	!=	"World"	INVALID
SECT112	中文	>	"World"	INVALID
SECT113	中文	<	"World"	INVALID
SECT114	中文	>=	"World"	INVALID
SECT115	中文	<=	"World"	INVALID
SECT116	中文	&	"World"	INVALID
SECT117	中文		"World"	INVALID
SECT118	中文	^	"World"	INVALID
SECT119	中文	~	"World"	INVALID
SECT120	中文	<<	"World"	INVALID

SECT121	中文	>>	"World"	INVALID
SECT122	中文	>>>	"World"	INVALID
SECT123	中文	&&	"World"	INVALID
SECT124	中文		"World"	INVALID
SECT125	中文	!	"World"	INVALID
SECT126	中文	=	"World"	INVALID
SECT127	中文	+=	"World"	INVALID
SECT128	中文	-=	"World"	INVALID
SECT129	中文	*=	"World"	INVALID
SECT130	中文	/=	"World"	INVALID
SECT131	中文	%=	"World"	INVALID
SECT132	中文	<<=	"World"	INVALID
SECT133	中文	>>=	"World"	INVALID
SECT134	中文	&=	"World"	INVALID
SECT135	中文	^=	"World"	INVALID
SECT136	中文	=	"World"	INVALID
SECT137	"Hello"	===	"World"	INVALID
SECT138	'c'	===	"World"	INVALID

SECT139	42	===	"World"	INVALID
SECT140	中文	===	"World"	INVALID
SECT141	"Hello"	+	'c'	VALID
SECT142	"Hello"	-	'c'	VALID
SECT143	"Hello"	*	'c'	VALID
SECT144	"Hello"	/	'c'	VALID
SECT145	"Hello"	%	'c'	VALID
SECT146	"Hello"	++	'c'	VALID
SECT147	"Hello"	--	'c'	VALID
SECT148	"Hello"	==	'c'	VALID
SECT149	"Hello"	!=	'c'	VALID
SECT150	"Hello"	>	'c'	VALID
SECT151	"Hello"	<	'c'	VALID
SECT152	"Hello"	>=	'c'	VALID
SECT153	"Hello"	<=	'c'	VALID
SECT154	"Hello"	&	'c'	VALID
SECT155	"Hello"		'c'	VALID
SECT156	"Hello"	^	'c'	VALID

SECT157	"Hello"	~	'c'	VALID
SECT158	"Hello"	<<	'c'	VALID
SECT159	"Hello"	>>	'c'	VALID
SECT160	"Hello"	>>>	'c'	VALID
SECT161	"Hello"	&&	'c'	VALID
SECT162	"Hello"		'c'	VALID
SECT163	"Hello"	!	'c'	VALID
SECT164	"Hello"	=	'c'	VALID
SECT165	"Hello"	+=	'c'	VALID
SECT166	"Hello"	-=	'c'	VALID
SECT167	"Hello"	*=	'c'	VALID
SECT168	"Hello"	/=	'c'	VALID
SECT169	"Hello"	%=	'c'	VALID
SECT170	"Hello"	<<=	'c'	VALID
SECT171	"Hello"	>>=	'c'	VALID
SECT172	"Hello"	&=	'c'	VALID
SECT173	"Hello"	^=	'c'	VALID
SECT174	"Hello"	=	'c'	VALID



SECT175	'c'	+	'c'	VALID
SECT176	'c'	-	'c'	VALID
SECT177	'c'	*	'c'	VALID
SECT178	'c'	/	'c'	VALID
SECT179	'c'	%	'c'	VALID
SECT180	'c'	++	'c'	VALID
SECT181	'c'	--	'c'	VALID
SECT182	'c'	==	'c'	VALID
SECT183	'c'	!=	'c'	VALID
SECT184	'c'	>	'c'	VALID
SECT185	'c'	<	'c'	VALID
SECT186	'c'	>=	'c'	VALID
SECT187	'c'	<=	'c'	VALID
SECT188	'c'	&	'c'	VALID
SECT189	'c'		'c'	VALID
SECT190	'c'	^	'c'	VALID
SECT191	'c'	~	'c'	VALID
SECT192	'c'	<<	'c'	VALID

SECT193	'c'	>>	'c'	VALID
SECT194	'c'	>>>	'c'	VALID
SECT195	'c'	&&	'c'	VALID
SECT196	'c'		'c'	VALID
SECT197	'c'	!	'c'	VALID
SECT198	'c'	=	'c'	VALID
SECT199	'c'	+=	'c'	VALID
SECT200	'c'	-=	'c'	VALID
SECT201	'c'	*=	'c'	VALID
SECT202	'c'	/=	'c'	VALID
SECT203	'c'	%=	'c'	VALID
SECT204	'c'	<<=	'c'	VALID
SECT205	'c'	>>=	'c'	VALID
SECT206	'c'	&=	'c'	VALID
SECT207	'c'	^=	'c'	VALID
SECT208	'c'	=	'c'	VALID
SECT209	42	+	'c'	VALID
SECT210	42	-	'c'	VALID

SECT211	42	*	'c'	VALID
SECT212	42	/	'c'	VALID
SECT213	42	%	'c'	VALID
SECT214	42	++	'c'	VALID
SECT215	42	--	'c'	VALID
SECT216	42	==	'c'	VALID
SECT217	42	!=	'c'	VALID
SECT218	42	>	'c'	VALID
SECT219	42	<	'c'	VALID
SECT220	42	>=	'c'	VALID
SECT221	42	<=	'c'	VALID
SECT222	42	&	'c'	VALID
SECT223	42		'c'	VALID
SECT224	42	^	'c'	VALID
SECT225	42	~	'c'	VALID
SECT226	42	<<	'c'	VALID
SECT227	42	>>	'c'	VALID
SECT228	42	>>>	'c'	VALID

SECT229	42	&&	'c'	VALID
SECT230	42		'c'	VALID
SECT231	42	!	'c'	VALID
SECT232	42	=	'c'	VALID
SECT233	42	+=	'c'	VALID
SECT234	42	-=	'c'	VALID
SECT235	42	*=	'c'	VALID
SECT236	42	/=	'c'	VALID
SECT237	42	%=	'c'	VALID
SECT238	42	<<=	'c'	VALID
SECT239	42	>>=	'c'	VALID
SECT240	42	&=	'c'	VALID
SECT241	42	^=	'c'	VALID
SECT242	42	=	'c'	VALID
SECT243	中文	+	'c'	INVALID
SECT244	中文	-	'c'	INVALID
SECT245	中文	*	'c'	INVALID
SECT246	中文	/	'c'	INVALID

SECT247	中文	%	'c'	INVALID
SECT248	中文	++	'c'	INVALID
SECT249	中文	--	'c'	INVALID
SECT250	中文	==	'c'	INVALID
SECT251	中文	!=	'c'	INVALID
SECT252	中文	>	'c'	INVALID
SECT253	中文	<	'c'	INVALID
SECT254	中文	>=	'c'	INVALID
SECT255	中文	<=	'c'	INVALID
SECT256	中文	&	'c'	INVALID
SECT257	中文		'c'	INVALID
SECT258	中文	^	'c'	INVALID
SECT259	中文	~	'c'	INVALID
SECT260	中文	<<	'c'	INVALID
SECT261	中文	>>	'c'	INVALID
SECT262	中文	>>>	'c'	INVALID
SECT263	中文	&&	'c'	INVALID
SECT264	中文		'c'	INVALID

SECT265	中文	!	'c'	INVALID
SECT266	中文	=	'c'	INVALID
SECT267	中文	+=	'c'	INVALID
SECT268	中文	-=	'c'	INVALID
SECT269	中文	*=	'c'	INVALID
SECT270	中文	/=	'c'	INVALID
SECT271	中文	%=	'c'	INVALID
SECT272	中文	<<=	'c'	INVALID
SECT273	中文	>>=	'c'	INVALID
SECT274	中文	&=	'c'	INVALID
SECT275	中文	^=	'c'	INVALID
SECT276	中文	=	'c'	INVALID
SECT277	"Hello"	===	'c'	INVALID
SECT278	'c'	===	'c'	INVALID
SECT279	42	===	'c'	INVALID
SECT280	中文	===	'c'	INVALID
SECT281	"Hello"	+	42	VALID
SECT282	"Hello"	-	42	VALID

SECT283	"Hello"	*	42	VALID
SECT284	"Hello"	/	42	VALID
SECT285	"Hello"	%	42	VALID
SECT286	"Hello"	++	42	VALID
SECT287	"Hello"	--	42	VALID
SECT288	"Hello"	==	42	VALID
SECT289	"Hello"	!=	42	VALID
SECT290	"Hello"	>	42	VALID
SECT291	"Hello"	<	42	VALID
SECT292	"Hello"	>=	42	VALID
SECT293	"Hello"	<=	42	VALID
SECT294	"Hello"	&	42	VALID
SECT295	"Hello"		42	VALID
SECT296	"Hello"	^	42	VALID
SECT297	"Hello"	~	42	VALID
SECT298	"Hello"	<<	42	VALID
SECT299	"Hello"	>>	42	VALID
SECT300	"Hello"	>>>	42	VALID

SECT301	"Hello"	&&	42	VALID
SECT302	"Hello"		42	VALID
SECT303	"Hello"	!	42	VALID
SECT304	"Hello"	=	42	VALID
SECT305	"Hello"	+=	42	VALID
SECT306	"Hello"	-=	42	VALID
SECT307	"Hello"	*=	42	VALID
SECT308	"Hello"	/=	42	VALID
SECT309	"Hello"	%=	42	VALID
SECT310	"Hello"	<<=	42	VALID
SECT311	"Hello"	>>=	42	VALID
SECT312	"Hello"	&=	42	VALID
SECT313	"Hello"	^=	42	VALID
SECT314	"Hello"	=	42	VALID
SECT315	'c'	+	42	VALID
SECT316	'c'	-	42	VALID
SECT317	'c'	*	42	VALID
SECT318	'c'	/	42	VALID



SECT319	'c'	%	42	VALID
SECT320	'c'	++	42	VALID
SECT321	'c'	--	42	VALID
SECT322	'c'	==	42	VALID
SECT323	'c'	!=	42	VALID
SECT324	'c'	>	42	VALID
SECT325	'c'	<	42	VALID
SECT326	'c'	>=	42	VALID
SECT327	'c'	<=	42	VALID
SECT328	'c'	&	42	VALID
SECT329	'c'		42	VALID
SECT330	'c'	^	42	VALID
SECT331	'c'	~	42	VALID
SECT332	'c'	<<	42	VALID
SECT333	'c'	>>	42	VALID
SECT334	'c'	>>>	42	VALID
SECT335	'c'	&&	42	VALID
SECT336	'c'		42	VALID

SECT337	'c'	!	42	VALID
SECT338	'c'	=	42	VALID
SECT339	'c'	+=	42	VALID
SECT340	'c'	-=	42	VALID
SECT341	'c'	*=	42	VALID
SECT342	'c'	/=	42	VALID
SECT343	'c'	%=	42	VALID
SECT344	'c'	<<=	42	VALID
SECT345	'c'	>>=	42	VALID
SECT346	'c'	&=	42	VALID
SECT347	'c'	^=	42	VALID
SECT348	'c'	=	42	VALID
SECT349	42	+	42	VALID
SECT350	42	-	42	VALID
SECT351	42	*	42	VALID
SECT352	42	/	42	VALID
SECT353	42	%	42	VALID
SECT354	42	++	42	VALID

SECT355	42	--	42	VALID
SECT356	42	==	42	VALID
SECT357	42	!=	42	VALID
SECT358	42	>	42	VALID
SECT359	42	<	42	VALID
SECT360	42	>=	42	VALID
SECT361	42	<=	42	VALID
SECT362	42	&	42	VALID
SECT363	42		42	VALID
SECT364	42	^	42	VALID
SECT365	42	~	42	VALID
SECT366	42	<<	42	VALID
SECT367	42	>>	42	VALID
SECT368	42	>>>	42	VALID
SECT369	42	&&	42	VALID
SECT370	42		42	VALID
SECT371	42	!	42	VALID
SECT372	42	=	42	VALID

SECT373	42	+=	42	VALID
SECT374	42	-=	42	VALID
SECT375	42	*=	42	VALID
SECT376	42	/=	42	VALID
SECT377	42	%=	42	VALID
SECT378	42	<<=	42	VALID
SECT379	42	>>=	42	VALID
SECT380	42	&=	42	VALID
SECT381	42	^=	42	VALID
SECT382	42	=	42	VALID
SECT383	中文	+	42	INVALID
SECT384	中文	-	42	INVALID
SECT385	中文	*	42	INVALID
SECT386	中文	/	42	INVALID
SECT387	中文	%	42	INVALID
SECT388	中文	++	42	INVALID
SECT389	中文	--	42	INVALID
SECT390	中文	==	42	INVALID

SECT391	中文	!=	42	INVALID
SECT392	中文	>	42	INVALID
SECT393	中文	<	42	INVALID
SECT394	中文	>=	42	INVALID
SECT395	中文	<=	42	INVALID
SECT396	中文	&	42	INVALID
SECT397	中文		42	INVALID
SECT398	中文	^	42	INVALID
SECT399	中文	~	42	INVALID
SECT400	中文	<<	42	INVALID
SECT401	中文	>>	42	INVALID
SECT402	中文	>>>	42	INVALID
SECT403	中文	&&	42	INVALID
SECT404	中文		42	INVALID
SECT405	中文	!	42	INVALID
SECT406	中文	=	42	INVALID
SECT407	中文	+=	42	INVALID
SECT408	中文	-=	42	INVALID

SECT409	中文	*=	42	INVALID
SECT410	中文	/=	42	INVALID
SECT411	中文	%=	42	INVALID
SECT412	中文	<<=	42	INVALID
SECT413	中文	>>=	42	INVALID
SECT414	中文	&=	42	INVALID
SECT415	中文	^=	42	INVALID
SECT416	中文	=	42	INVALID
SECT417	"Hello"	===	42	INVALID
SECT418	'c'	===	42	INVALID
SECT419	42	===	42	INVALID
SECT420	中文	===	42	INVALID
SECT421	"Hello"	+	中文	INVALID
SECT422	"Hello"	-	中文	INVALID
SECT423	"Hello"	*	中文	INVALID
SECT424	"Hello"	/	中文	INVALID
SECT425	"Hello"	%	中文	INVALID
SECT426	"Hello"	++	中文	INVALID

SECT427	"Hello"	--	中文	INVALID
SECT428	"Hello"	==	中文	INVALID
SECT429	"Hello"	!=	中文	INVALID
SECT430	"Hello"	>	中文	INVALID
SECT431	"Hello"	<	中文	INVALID
SECT432	"Hello"	>=	中文	INVALID
SECT433	"Hello"	<=	中文	INVALID
SECT434	"Hello"	&	中文	INVALID
SECT435	"Hello"		中文	INVALID
SECT436	"Hello"	^	中文	INVALID
SECT437	"Hello"	~	中文	INVALID
SECT438	"Hello"	<<	中文	INVALID
SECT439	"Hello"	>>	中文	INVALID
SECT440	"Hello"	>>>	中文	INVALID
SECT441	"Hello"	&&	中文	INVALID
SECT442	"Hello"		中文	INVALID
SECT443	"Hello"	!	中文	INVALID
SECT444	"Hello"	=	中文	INVALID

SECT445	"Hello"	+=	中文	INVALID
SECT446	"Hello"	-=	中文	INVALID
SECT447	"Hello"	*=	中文	INVALID
SECT448	"Hello"	/=	中文	INVALID
SECT449	"Hello"	%=	中文	INVALID
SECT450	"Hello"	<<=	中文	INVALID
SECT451	"Hello"	>>=	中文	INVALID
SECT452	"Hello"	&=	中文	INVALID
SECT453	"Hello"	^=	中文	INVALID
SECT454	"Hello"	=	中文	INVALID
SECT455	'c'	+	中文	INVALID
SECT456	'c'	-	中文	INVALID
SECT457	'c'	*	中文	INVALID
SECT458	'c'	/	中文	INVALID
SECT459	'c'	%	中文	INVALID
SECT460	'c'	++	中文	INVALID
SECT461	'c'	--	中文	INVALID
SECT462	'c'	==	中文	INVALID



SECT463	'c'	!=	中文	INVALID
SECT464	'c'	>	中文	INVALID
SECT465	'c'	<	中文	INVALID
SECT466	'c'	>=	中文	INVALID
SECT467	'c'	<=	中文	INVALID
SECT468	'c'	&	中文	INVALID
SECT469	'c'		中文	INVALID
SECT470	'c'	^	中文	INVALID
SECT471	'c'	~	中文	INVALID
SECT472	'c'	<<	中文	INVALID
SECT473	'c'	>>	中文	INVALID
SECT474	'c'	>>>	中文	INVALID
SECT475	'c'	&&	中文	INVALID
SECT476	'c'		中文	INVALID
SECT477	'c'	!	中文	INVALID
SECT478	'c'	=	中文	INVALID
SECT479	'c'	+=	中文	INVALID
SECT480	'c'	-=	中文	INVALID

SECT481	'c'	*=	中文	INVALID
SECT482	'c'	/=	中文	INVALID
SECT483	'c'	%=	中文	INVALID
SECT484	'c'	<<=	中文	INVALID
SECT485	'c'	>>=	中文	INVALID
SECT486	'c'	&=	中文	INVALID
SECT487	'c'	^=	中文	INVALID
SECT488	'c'	=	中文	INVALID
SECT489	42	+	中文	INVALID
SECT490	42	-	中文	INVALID
SECT491	42	*	中文	INVALID
SECT492	42	/	中文	INVALID
SECT493	42	%	中文	INVALID
SECT494	42	++	中文	INVALID
SECT495	42	--	中文	INVALID
SECT496	42	==	中文	INVALID
SECT497	42	!=	中文	INVALID
SECT498	42	>	中文	INVALID

SECT499	42	<	中文	INVALID
SECT500	42	>=	中文	INVALID
SECT501	42	<=	中文	INVALID
SECT502	42	&	中文	INVALID
SECT503	42		中文	INVALID
SECT504	42	^	中文	INVALID
SECT505	42	~	中文	INVALID
SECT506	42	<<	中文	INVALID
SECT507	42	>>	中文	INVALID
SECT508	42	>>>	中文	INVALID
SECT509	42	&&	中文	INVALID
SECT510	42		中文	INVALID
SECT511	42	!	中文	INVALID
SECT512	42	=	中文	INVALID
SECT513	42	+=	中文	INVALID
SECT514	42	-=	中文	INVALID
SECT515	42	*=	中文	INVALID
SECT516	42	/=	中文	INVALID

SECT517	42	%=	中文	INVALID
SECT518	42	<<=	中文	INVALID
SECT519	42	>>=	中文	INVALID
SECT520	42	&=	中文	INVALID
SECT521	42	^=	中文	INVALID
SECT522	42	=	中文	INVALID
SECT523	中文	+	中文	INVALID
SECT524	中文	-	中文	INVALID
SECT525	中文	*	中文	INVALID
SECT526	中文	/	中文	INVALID
SECT527	中文	%	中文	INVALID
SECT528	中文	++	中文	INVALID
SECT529	中文	--	中文	INVALID
SECT530	中文	==	中文	INVALID
SECT531	中文	!=	中文	INVALID
SECT532	中文	>	中文	INVALID
SECT533	中文	<	中文	INVALID
SECT534	中文	>=	中文	INVALID

SECT535	中文	<=	中文	INVALID
SECT536	中文	&	中文	INVALID
SECT537	中文		中文	INVALID
SECT538	中文	^	中文	INVALID
SECT539	中文	~	中文	INVALID
SECT540	中文	<<	中文	INVALID
SECT541	中文	>>	中文	INVALID
SECT542	中文	>>>	中文	INVALID
SECT543	中文	&&	中文	INVALID
SECT544	中文		中文	INVALID
SECT545	中文	!	中文	INVALID
SECT546	中文	=	中文	INVALID
SECT547	中文	+=	中文	INVALID
SECT548	中文	-=	中文	INVALID
SECT549	中文	*=	中文	INVALID
SECT550	中文	/=	中文	INVALID
SECT551	中文	%=	中文	INVALID
SECT552	中文	<<=	中文	INVALID

SECT553	中文	>>=	中文	INVALID
SECT554	中文	&=	中文	INVALID
SECT555	中文	^=	中文	INVALID
SECT556	中文	=	中文	INVALID
SECT557	"Hello"	===	中文	INVALID
SECT558	'c'	===	中文	INVALID
SECT559	42	===	中文	INVALID
SECT560	中文	===	中文	INVALID

**White Box:** Control Flow Graph, c-use, p-use, def-use, all-uses criterion.

## Control Flow Graph

### Operator function:

NumberOfOperatorCheck.java

```
1 package net.sf.eclipsecs.sample.checks;
2
3 import java.util.HashMap;
4
5
6
7
8
9
10 public class NumberOfOperatorCheck extends AbstractCheck {
11     private int totalOperator = 0;
12     private int unique_operator = 0;
13     private Map<Integer, Integer> map = new HashMap<Integer, Integer>();
14
15     public int[] getDefaultTokens() {
16         return new int[] { TokenTypes.CLASS_DEF, TokenTypes.INTERFACE_DEF };
17     }
18
19     @Override
20     public void visitToken(DetailAST ast) {
21         totalOperator = this.getTotalOperator(ast);
22         System.out.println("operator: " + totalOperator);
23         unique_operator = this.getUniqueOperator(ast);
24         System.out.println("unique operator: " + unique_operator);
25         log(ast.getLineNo(), "numberOfOperator", totalOperator/2);
26         log(ast.getLineNo(), "uniqueOfOperator", unique_operator);
27     }
28
29     /*
30      * getter for total Operator
31      */
32     public int getTotalOperator(DetailAST cur) {
33         totalOperator = 0;
34         map = new HashMap<Integer, Integer>();
35         recFindOperator(cur);
36         return this.totalOperator;
37     }
38
39     /*
40      * getter for total Operator
41      */
42     public int getUniqueOperator(DetailAST cur) {
43         unique_operator = 0;
44         map = new HashMap<Integer, Integer>();
45         recFindOperator(cur);
46         return this.map.size();
47     }
48
49     //using recursion to traverse ever node
50     public void recFindOperator(DetailAST cur) {
51         if (cur == null) {
52             return;
53         }
54         if (cur.getType() == TokenTypes.ASSIGN || cur.getType() == TokenTypes.BAND
55             || cur.getType() == TokenTypes.BAND_ASSIGN || cur.getType() == TokenTypes.BNOT
56             || cur.getType() == TokenTypes.BOR || cur.getType() == TokenTypes.BOR_ASSIGN
57             || cur.getType() == TokenTypes.BSR || cur.getType() == TokenTypes.BSR_ASSIGN
58             || cur.getType() == TokenTypes.BXOR || cur.getType() == TokenTypes.BXOR_ASSIGN
59             || cur.getType() == TokenTypes.COLON || cur.getType() == TokenTypes.COMMA
60             || cur.getType() == TokenTypes.DEC || cur.getType() == TokenTypes.DIV
61             || cur.getType() == TokenTypes.DIV_ASSIGN || cur.getType() == TokenTypes.DOT
62             || cur.getType() == TokenTypes.EQUAL || cur.getType() == TokenTypes.GE ||
```

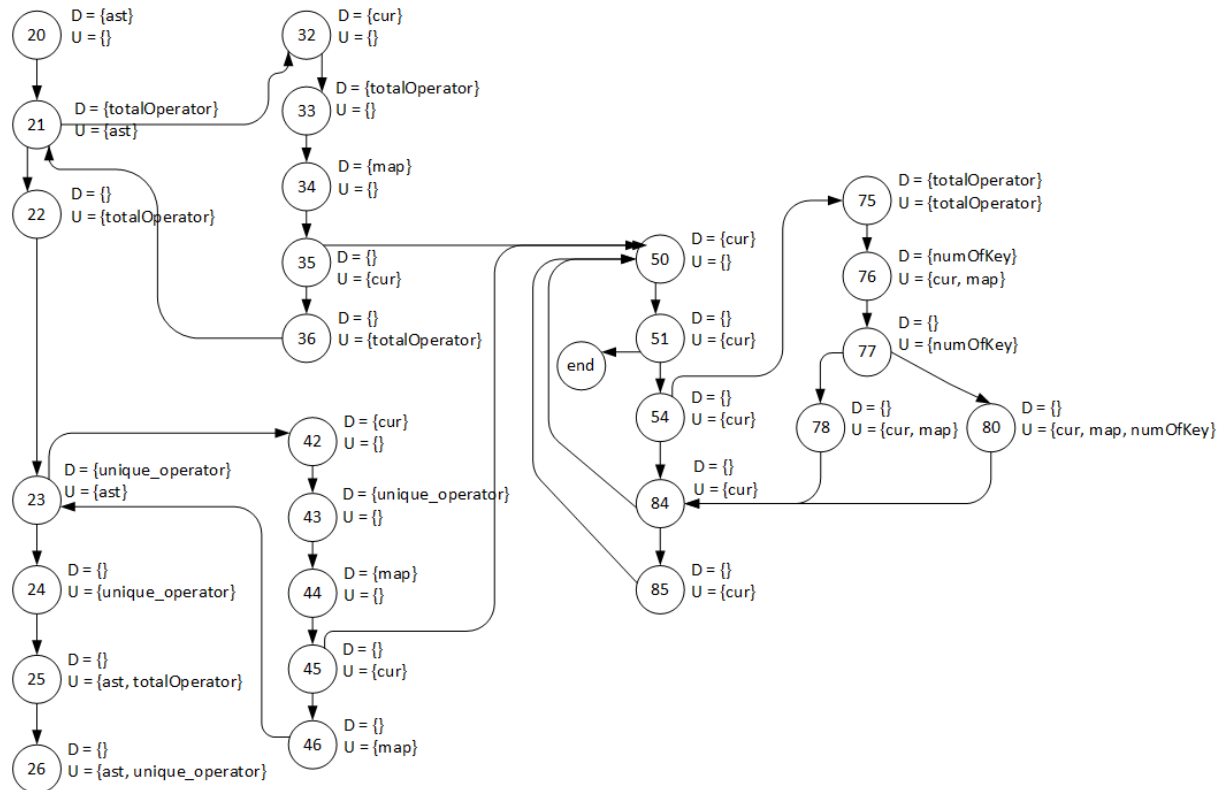
# NumberOfOperatorCheck.java

```

cur.getType() == TokenTypes.GT
63      || cur.getType() == TokenTypes.INC || cur.getType() == TokenTypes.INDEX_OP
64      || cur.getType() == TokenTypes.LAND || cur.getType() ==
TokenTypes.LITERAL_INSTANCEOF
65      || cur.getType() == TokenTypes.LNOT || cur.getType() == TokenTypes.LOR ||
cur.getType() == TokenTypes.LT
66      || cur.getType() == TokenTypes.MINUS || cur.getType() ==
TokenTypes.MINUS_ASSIGN
67      || cur.getType() == TokenTypes.MOD || cur.getType() == TokenTypes.MOD_ASSIGN
68      || cur.getType() == TokenTypes.NOT_EQUAL || cur.getType() == TokenTypes.PLUS
69      || cur.getType() == TokenTypes.PLUS_ASSIGN || cur.getType() ==
TokenTypes.POST_DEC
70      || cur.getType() == TokenTypes.POST_INC || cur.getType() == TokenTypes.QUESTION
71      || cur.getType() == TokenTypes.SL || cur.getType() == TokenTypes.SL_ASSIGN
72      || cur.getType() == TokenTypes.SR || cur.getType() == TokenTypes.SR_ASSIGN
73      || cur.getType() == TokenTypes.STAR || cur.getType() == TokenTypes.STAR_ASSIGN
74      || cur.getType() == TokenTypes.UNARY_MINUS || cur.getType() ==
TokenTypes.UNARY_PLUS) {
75      this.totalOperator++;
76      Object numOfKey = map.get(cur.getType());
77      if (numOfKey == null) {
78          map.put(cur.getType(), 1);
79      } else {
80          map.put(cur.getType(), (int) numOfKey + 1);
81      }
82  }
83
84      recFindOperator(cur.getFirstChild());
85      recFindOperator(cur.getNextSibling());
86  }
87 }
88

```

## Operator CFG:





(Graph start at line 20, code before line 20 are global definition)

**Provide all c-uses and p-uses for operator function.**

[illegible]

34												
35							X					
36			X									
37												
38												
39												
40												
41												
42												
43												
44												
45							X					
46									X			
47												
48												
49												
50												
51								X				

52												
53												
54							X	X				
55							X	X				
56							X	X				
57							X	X				
58							X	X				
59							X	X				
60							X	X				
61							X	X				
62							X	X				
63							X	X				
64							X	X				
65							X	X				
66							X	X				
67							X	X				
68							X	X				
69							X	X				

[illegible]

**Identify all def-uses for all variables for operator function.**

	Definition					
Line	ast	totalOperator	unique_operator	cur	map	numOfKey
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						

16						
17						
18						
19						
20	{21,23,25,26}					
21		{22,25}				
22						
23			{24,26}			
24						
25						
26						
27						
28						
29						
30						
31						
32				{35}		
33		{36,75}				

34					{76,78,80}	
35						
36						
37						
38						
39						
40						
41						
42				{45}		
43						
44					{46,76,78,80}	
45						
46						
47						
48						
49						
50				{51,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74}		

				,76,78,80,84,85}		
51						
52						
53						
54						
55						
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						



68						
69						
70						
71						
72						
73						
74						
75		{36,75}				
76						{77,80}
77						
78						
79						
80						
81						
82						
83						
84						
85						

86						
87						

**Provide test cases which satisfy the all-uses criterion for operator function.**

Test case	ast	DU covered					
		ast	totalOperator	unique_operator	cur	map	numOfKey
T1	<pre> public class hw {     public static void     main(String[] args) {         int int_num = 0;         double double_nul =         0.0;         string str;         Obj o1 = new         Obj(1);         Obj o2 = new         Obj(2);         o1 = o2;         char c = 65;          System.out.println(c);     } } class Obj {     int id;      Obj(int _id)     {         id = _id;     } } </pre>	<pre> &lt;20,21&gt; &lt;20,23&gt; &lt;20,25&gt; &lt;20,26&gt; </pre>	<pre> &lt;21,22&gt; &lt;21,25&gt; &lt;33,36&gt; &lt;33,75&gt; &lt;75,36&gt; &lt;75,75&gt; </pre>	<pre> &lt;23,24&gt; &lt;23,26&gt; </pre>	<pre> &lt;32,35&gt; &lt;42,45&gt; &lt;50,54&gt; &lt;50,55&gt; &lt;50,56&gt; &lt;50,57&gt; &lt;50,58&gt; &lt;50,59&gt; &lt;50,60&gt; &lt;50,61&gt; &lt;50,62&gt; &lt;50,63&gt; &lt;50,64&gt; &lt;50,65&gt; &lt;50,66&gt; &lt;50,67&gt; &lt;50,68&gt; &lt;50,69&gt; &lt;50,70&gt; &lt;50,71&gt; &lt;50,72&gt; &lt;50,73&gt; &lt;50,74&gt; &lt;50,76&gt; &lt;50,78&gt; &lt;50,80&gt; &lt;50,84&gt; &lt;50,85&gt; </pre>	<pre> &lt;34,76&gt; &lt;34,78&gt; &lt;34,80&gt; &lt;44,46&gt; &lt;44,76&gt; &lt;44,78&gt; &lt;44,80&gt; </pre>	<pre> &lt;76,77&gt; &lt;76,80&gt; </pre>

## Operand function:

NumberOfOperandCheck.java

```
1 package net.sf.eclipsecs.sample.checks;
2
3 import java.util.HashMap;
4
5
6
7
8
9
10
11 public class NumberOfOperandCheck extends AbstractCheck {
12     private int totalOperand = 0;
13     private int unique_operand = 0;
14     private Map<String, Integer> map = new HashMap<String, Integer>();
15
16     public int[] getDefaultTokens() {
17         return new int[] { TokenTypes.CLASS_DEF, TokenTypes.INTERFACE_DEF };
18     }
19
20     @Override
21     public void visitToken(DetailAST ast) {
22         totalOperand = this.getTotalOperand(ast);
23         System.out.println("operand: " + totalOperand);
24
25         unique_operand = this.getUniqueOperand(ast);
26         System.out.println("unique operand: " + unique_operand);
27
28         log(ast.getLineNo(), "numberOfOperand", totalOperand/2);
29         log(ast.getLineNo(), "uniqueOfOperand", unique_operand);
30     }
31
32     /*
33     * getter for total operand
34     */
35     public int getTotalOperand(DetailAST cur)
36     {
37         totalOperand = 0;
38         map = new HashMap<String, Integer>();
39         recFindOperand(cur);
40         return this.totalOperand;
41     }
42
43     /*
44     * getter for unique operand
45     */
46     public int getUniqueOperand(DetailAST cur)
47     {
48         unique_operand = 0;
49         map = new HashMap<String, Integer>();
50
51         recFindOperand(cur);
52         return this.map.size();
53     }
54
55     //using recursion to traverse ever node
56     public void recFindOperand(DetailAST cur) {
57         if (cur == null) {
58             return;
59         }
60         //cur.getType() == TokenTypes.IDENT
61         if (cur.getType() == TokenTypes.NUM_INT || cur.getType() == TokenTypes.NUM_FLOAT ||
62             cur.getType() == TokenTypes.IDENT) {
```

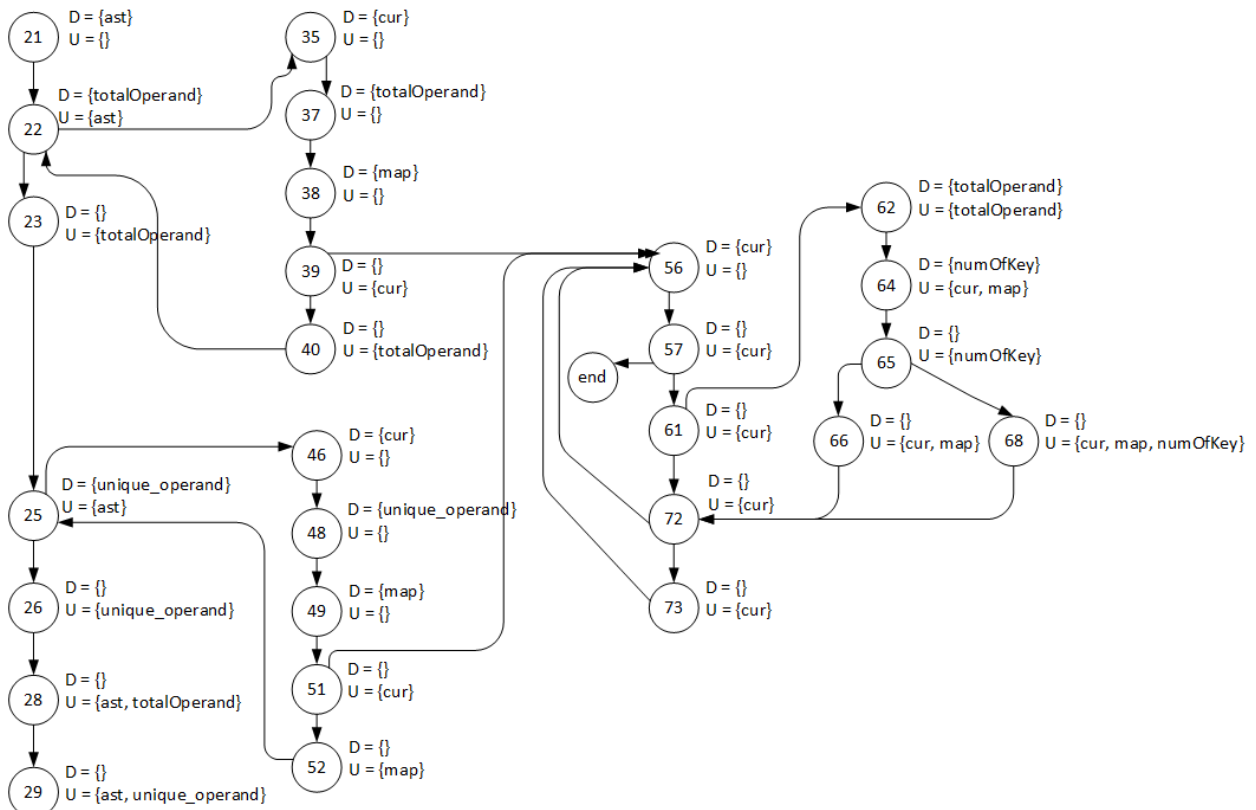
# NumberOfOperandCheck.java

```

62         this.totalOperand++;
63
64         Object numOfKey = map.get(cur.getText());
65         if (numOfKey == null) {
66             map.put(cur.getText(), 1);
67         } else {
68             map.put(cur.getText(), (int) numOfKey + 1);
69         }
70     }
71
72     recFindOperand(cur.getFirstChild());
73     recFindOperand(cur.getNextSibling());
74 }
75 }
76

```

## Operand CFG:



(Graph start at line 21, code before line 21 are global definition)

**Provide all c-uses and p-uses for operand function.**

[illegible]

34												
35												
36												
37												
38												
39							X					
40			X									
41												
42												
43												
44												
45												
46												
47												
48												
49												
50												
51							X					

[illegible]

70												
71												
72							X					
73							X					
74												
75												

**Identify all def-uses for all variables for operand function.**

	Definition					
Line	ast	totalOperand	unique_operand	cur	map	numOfKey
1						
2						
3						
4						
5						
6						
7						
8						
9						



10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21	{22,25,28,29}					
22		{23,28}				
23						
24						
25			{26,29}			
26						
27						

28						
29						
30						
31						
32						
33						
34						
35				{39}		
36						
37		{40, 62}				
38					{64,66,68}	
39						
40						
41						
42						
43						
44						
45						

46				{51}		
47						
48						
49					{52,64,6 6,68}	
50						
51						
52						
53						
54						
55						
56				{57,61,64,6 6,68,72,73}		
57						
58						
59						
60						
61						
62						

63						
64						{65,68}
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						

**Provide test cases which satisfy the all-uses criterion for operand function.**

Test case	ast	DU covered					
		ast	totalOper and	unique_ope rand	cur	map	numOfKey

T1	<pre> public class hw {      public static void main(String[] args) {     int int_num = 0;     double double_nul = 0.0;     string str;     Obj o1 = new Obj(1);     Obj o2 = new Obj(2);     o1 = o2;     char c = 65;      System.out.println(c );     } } class Obj {     int id;      Obj(int _id)     {         id = _id; }} </pre>	<21,22>	<22,23>	<25,26>	<35,39>	<38,64>	<64,65>
		<21,25>	<22,28>	<25,29>	<46,51>	<38,66>	<64,68>
		<21,28>	<37,40>		<56,57>	<38,68>	
		<21,29>	<37,62>		<56,61>	<49,52>	
					<56,64>	<49,64>	
					<56,66>	<49,66>	
					<56,68>	<49,68>	
					<56,72>		
					<56,73>		