	Hash Map :>
	The vorter line of data stoueterre for Hammap 1s Hash Table,
	insurfor order is not preserved and it is based on
	hashtode of keys.
	puplicate Kys are not allowed but values can be
رني ري	duplicated.
	lectorgenions objects are allowed for both key and Value.
2	Null is allowed for key (only once),
	Null is allowed for values (any mos of times)
77	It implements Sevializable, and Cloneable interfece but not
	Rando on Access.
8	HashMap is the best Choice if our frequent open is search.
	constructors! - and vien
	The second of th
	HashMap ron = new HashMap ();
	creates an empty hash Malp Object with default
	initial capacity 16 and Default feel ratio is 0.75.
(2)	HOLMAN TO - new HashMap (int initial (apacity);
	empty HashMab Objects with Specified
	Pritial capacity and default feel ratio \$ 0.75.
	1 / Cal with Man FleRation
(3)	HashMap m= new HashMap (int initial Capacity, float fileRatio):
4	HashMap m = new HashMap (Map m);
	import Java. util. *)
	Test 13
	public class to m (Stronger)
	A
,	HashMap m = new HoshMap ();
	The state of the s
	mobut ("venkateth", 200); mi put (raggi juin)
	A LANGE OF STATE OF S
	$S \circ b(m) \Rightarrow \{ k = 1, 1, \dots, 1 \}$ Scanned by CamScanner
	Scanned by Camscanner

Set s: m. key Set (); Set s: m. key Set (); So: p(s); \(\to \) [k, K, K \(\to \) \\ So: p(c); Set s: m. entry Set(); Map. Entry m: [Map. Entry] itr. next (); So: p(m: get key() f! " + m. ext of control o	
Set s: m. key Sit () So () (s); \(\) [k, \(\) (\(\) () \) Set siz m. entry Sit (); So () (s)); \(\) [k = \(\), \(\) \(\) (); Set siz m. entry Sit (); So () (si); \(\) [k = \(\), \(\) \(\) (); Therator ftr; si. iterator (); While (ltr. has Next ()) [Map. Entry mi; (Map. Entry) ftr. next (); So () (mi. get Key () f! - " + mi. get Value (); 'f (mi. get Key (). equals ("nagar Juna")) [Mi. set Value (10000); \] [Mi. set Value (10000); \] So () (mi. get Key () equals ("nagar Juna")) [Mi. set Value (10000); \] [Mi. set Value (10000); Not Thread set [Mi. set Value (10000); Mot Thread set [Mi. set Value (10000); [Mi. set Value (10000); Mot Thread set [Mi. set Value (10000); [Map. Entry () 1	sup mi but ("chiranjivi", 1000))
S.O. p (S); \$\frac{1}{2} \ k, K, K, \frac{1}{2} \\ \text{(ollection c = movalues();} \\ \text{Set S12 m. entry Set()} \\ \text{So.p(S1)} \times \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	
Solp(c); Set sizm. entry Cut(); Solp(si) > E K=V, K=V,	$(n)h(s)^{*} \Rightarrow \lceil k, \kappa, \kappa - r \rceil$
S. o. p (c); Set \$12 m. entry Set(); \$0. p(\$1); \rightarrow [K=V, K=V, \rightarrow] Iterator itr=\$1. iterator (); White (Itr. has Next ()) Map. Entry m1= (Map. Entry) itr. next (); \$0. p (m1. get Key () + "" + m1. get Value (); if (m1. get Key (). equals ("nagar Juna")) \$m1. set Value (10000); \$m1. set Value (10000); \$m1. set Value (10000); The hash Map & Hash Table (methods one Synchronised one on Synchronised (methods one Synchronised one on Synchronised one on Synchronised (methods one on Synchronised one on Synchronised one on Synchronised one on Synchronised (methods one on Synchronised on one of Synchronised on	Collection (= movalues ();
Set \$1= m. entry (atl); \$00 p(\$1)' \rightarrow [K=V, K=V,] Therator itr = \$1. iterator ()' While (itr. has Next ()) [Map. Entry m1= (Map. Entry) itr. next (); \$00 p (m1. get Key () f" - "" + m1. get Value ()'; if (m1. get Key (). equals ("nagar Juna")) [Map. Entry m1= (Map. Entry) itr. next (); \$00 p (m1. get Key () f" - "" + m1. get Value ()'; if (m1. get Key () for a gar Juna")) [Map. Entry m1= (Map. Entry) itr. next (); m1. get Value ()'; if (m1. get Key () f" - "" + m1. get Value ()'; if (m1. get Key () f" magar Juna")) [Map. Entry m1= (Map. Entry) itr. next (); m1. get Value (10060); [Map. Entry m1= (Map. Entry) itr. next (); m1. get Value () for a gar Juna") [Map. Entry m1= (Map. Entry) itr. next (); m1. get Value (10060); [Map. Entry m1= (Map. Entry) itr. next (); m1. get Value (10060); [Map. Entry m1= (Map. Entry) itr. next (); m1. get Value (10060); [Map. Entry m1= (Map. Entry) itr. next (); m1. get Value (10060); [Map. Entry m2 (1006); m1. get Value (1006); m1. get Value (10060); m1. get Value (1006	
S.o.p(s1); > [K=V, K=V, r] Iterator itr=s1. iterator (); while (itr.hasNext ()) Map. Entry m1= (Map. Entry) itr.next (); So. p (m1. get Key () + "-" + m1. get Value ()); if (m1. get Key (). equals ("nagar Juna")) M1. set Value (10000); S.o.p(m); S.o.p(m); Methods are non Synchronized () Methods are Synchronized () Mot Thread safe () Thread safe () Mull is allowed as key () Null is not allowed as and value. (E:- NPE Shot Legacy (1.2 V) (5) Legacy (1.0 V)	Set C12 m; entry Set()/
Therator Ar = 51. lerator (); while (ltr. has Next ()) (Map. Entry m1 = (Map. Entry) itr. next (); 50. p (m1. get Key () + " - " + m1. get Value (); if (m1. get Key () equals ("nagar Juna")) Sop p (m); Sop p (m); Diff blw HashMap & HashTable :> HashMap HashTable () methods are non synchronized () Nethoods are Synchronized () Not Thread safe () thread safe () thread safe () For promatice is high () performance is low. (A) Null is allowed as key () Null is not allowed as and value, (CE: NPE	$Suph(SI)$, $\rightarrow \Gamma K=V, K=V, \uparrow \uparrow \uparrow$
while (Itr. has Next ()) (Map. Entry m1; (Map. Entry) itr. next (); So. p (m1. get Key () f " " + m1. get Value (); if (m1. get Key (). equals ("nagar Juna")) Sm1. set Value (10000); 3 S.O. p (m); Diff b w Hashmap & HashTable :> Hashmap HashTable () methods are non synchronized () Nethods are Synchronized () not Thread safe () Thread safe () terformatice is high () performance is 10 w. () Null is allowed as key () Null is not allowed as and value, (ce: - NPE (S) Not legacy (1.2 V) (S) Legacy (1.0 V)	Therator itrasi iterator ();
Map. Entry m1= (Map. Entry) itr. next (); So. p (m1. get Key () f " - " + m1. get Value ()); if (m1. get Key (). equals ("nagar Juna")) Sm1. set Value (10000); 3 S.O. p (m); Hash Map Hash Table O method are non synchronized () Nethods orce Synchronized Not Thread safe (2) Thread safe ferformatice is high (3) performance is low. Mull is allowed as key (1) Null is not allowed as and value. (E! - NPE (5) Not Legacy (1.2 V) S Legacy (1.0 V)	while 1 ltr. has Next ())
50. b (m1. get Key () f " - " + m1. get Value ()); if (m1. get Key () . equals ("nagar Juna")) Sm1. set Value (10000); 3 S.o. p (m); Hash Map B Hash Table :> Hash Map Hash Table O methods are non synchronized () Nethods are Synchronized Not Thread safe (2) Thread safe B herformate ce is high (3) performance is low. Mull is allowed as key (1) Null is not allowed as and value. Shot legacy (1.2 V) S legacy (1.0 V)	1
50. p (m1, get Key () f " - " + m1. get Value () of (nagar Juna")) f (m1. get Key (). equals ("nagar Juna")) m1. set Value (10000); 3 S.o. p (m); G.o. p (m); Hash Map Hash Table :> Hash Map Hash Table methods are non synchronized () Nethods are Synchronized methods are non synchronized () Nethods are synchronized () Nethods are synchronized () Nethods are synchronized () Nethods are synchronized () Nethods ar	Mah Entry mi - (Mab. Fortry) itronext ();
mi.get value (1); if (mi.get Key ().equals ("nagar Juna")) Smi.set Value (10000); G.o.p (m); Shot Legacy (1.2 V) If (mi.get Key ().equals ("nagar Juna")) Mi.set Value (10000); Mi.set Value (10000); Magar Juna") Methodo of; Methodo ore Synchronized Thread safe Thread safe Mull is allowed as key Mull is not allowed as CE!—NPE	50 h 1 m1, 90x Key () + " "+
m1. setValue (10000); S. 0. p m); S. 0. p m); Diff blw HashMap 8 HashTable :> HashMap HashTable O methods are non Synchroniad (1) Nethools are Synchroniad Not Thread safe (2) Thread safe Ferformatice is high (3) performance is low. Mull is allowed as key (1) Null is not allowed as and value. Shot Legacy (1.2 v) Elegacy (1.0 v)	
m1. setValue (10000); 3 S.o.p(m); Diff blw HashMap & HashTable :> HashMap HashTable O methods are non Synchronized (1) Methods one Synchronized D Not Thread safe (2) Thread safe (3) performance is high (3) performance is low. Mull is allowed as key (1) Null is not allowed as and value. Shot Legacy (1.2 V) S Legacy (1.0 V)	if (mi. get Key (). equals ("nagar Juna"))
S.O'p(m); S.O'p(m); Diff blw HashMap & HashTable; HashMap HashTable O methods are non Synchronized () Methods are Synchronized Not Thread Safe (2) Thread Safe B herformatice is high (3) performance is low. Mull is allowed as key (1) Null is not allowed as and value. Shot legacy (1.2 V) Shot legacy (1.2 V) Shot legacy (1.0 V)	5
S.O'p(m); S.O'p(m); Diff blw HashMap & HashTable; HashMap HashTable O methods are non Synchronized () Methods are Synchronized Not Thread Safe (2) Thread Safe B herformatice is high (3) performance is low. Mull is allowed as key (1) Null is not allowed as and value. Shot legacy (1.2 V) Shot legacy (1.2 V) Shot legacy (1.0 V)	m1. set/value (10000);
S.O.p(m); Diff blw HashMap & HashTable: HashMap HashTable O methodo are non Synchronizal (i) Methodo are Synchronizal Not Thread Safe (ii) Thread Safe Briformatice is high (ii) performance is low. P Null is allowed as key (iii) Null is not allowed as and value. CE:—NPE SNot Legacy (1.2 V) Equacy (1.0 V)	4
Diff blw HashMap & HashTable: > HashMap HashTable O methods are non Synchronized (i) Methods are Synchronized D Not Thread safe (ii) Thread safe Ferformatice is high (ii) performance is low. P Null is allowed as key (iii) Null is not allowed as and value. CE! — NPE Shot Legacy (1.2 V) (ii) Legacy (1.0 V)	
The short of the state of the s	
The short of the state of the s	
The shoot are non synchronized (i) Methods are Synchronized Derformatice is high (i) performance is low. Phull is allowed as key (i) Null is not allowed as and value (ii) and value (iii) and value (iii) (iii) and value (iii) (iii) and value (iii) (iii) and value (iiii) (iii) and value (iiii) (iii) and value (iii) (iii) and value (iii) (iii) and value (iii) (ii	= Diff blw HashMap & HashTable :>
Methods are non synchronized (1) Methods are Synchronized (3) Not Thread safe (2) Thread safe (3) performance is low. (4) Null is allowed as key (4) Null is not allowed as an and value. (5) Not legacy (1.2 V) (5) Legacy (1.0 V)	Hash Mab
3 performatice is high 3 performance is low. Thui is allowed as key 10 NWI is not allowed as and value. Shot legacy (1.2 V) 3 legacy (1.0 V)	methods are non synchronized (1) Methods are C
Thul is allowed as key Mull is not allowed as and value. Shot legacy (1.2 V) B legacy (1.0 V)	(bread sale
and value. Shot legacy (1.2 V) B legacy (1.0 V)	2) performatice is high (3) berformance is to
SNot legacy (1.2 V) B Legacy (1.0 V)	13 WIDWED OF KEY (A) NILL !
(5) Not legacy (1.2 V) (E! - NPE (1.2 V) (1.0 V)	Key and Value
	Blegary (1.2 V)
- 1 How to get Synchronia d march	
Cally to a last of Hashmab Abrest 1.	get Synchronized Wersian of
is good supplied to	is grant prashmap is non suchama
That Mah hall (1.12)	That Mah hu ()
Synchronized Map () method of Collections class.	Synchronized Map () method of Collections class.