```
String s = new String("hi");
String s1 = "hello"; // declaration of string.
```

taking user input

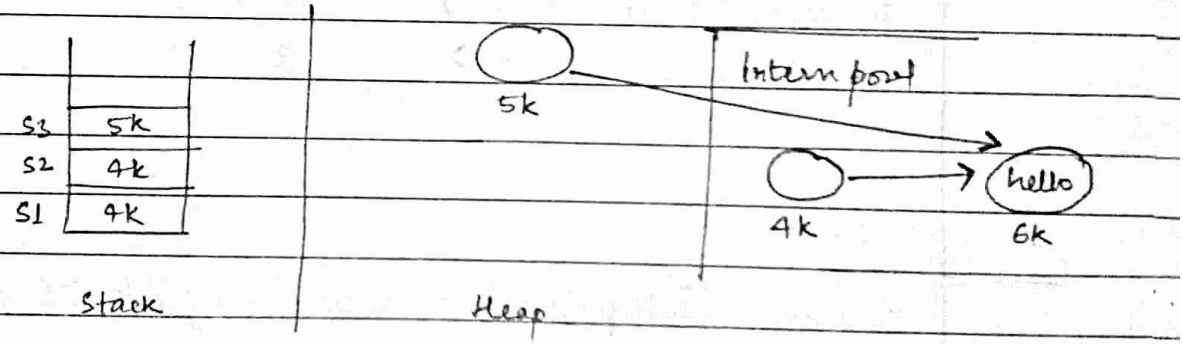
```
String s2 = sc.next(); // read till first space
String s3 = sc.nextLine(); // read whole line

char ch = str.charAt(0); // reading character.

String str = "abc def ghi";
String[] parts = str.split(" "); // split on space.
```

Memory & Interning — to optimize space

```
String s1 = "hello"; ✓
String s2 = "hello"; ✓
String s3 = new String("hello"); ✓
```



Equals & ==

Equals → Check character by character.

== → check address than content

So, `s1 == s2` // true — due to same addr
`s1 == s3` // false — due to diff addr.

∴ don't compare using ==

and `s1.equals(s2)` // true
`s1.equals(s3)` // true

* Immutability of Strings :-

Once a string object is created, its data or state can't be changed but the string can point to another character array.

Mutable String : StringBuilder

```
StringBuilder sb = new StringBuilder("hello");

char ch = sb.charAt(0);          // get      → h
sb.setCharAt(0, 'd');            // update   → dello
sb.insert(2, 'y');               →  deyllo
sb.deleteCharAt(2);              →  dello
sb.append("ite");                →  delloite
sb.length();                     →  8
```

Arraylists

```
ArrayList<Integer> list = new ArrayList<>();

list.add(10); list(20); list.add(30);  → [10, 20, 30]
list.add(1, 1000);                     → [10, 1000, 20, 30]
int val = list.get(1);                 → 1000
list.set(1, 2000);                     → [10, 2000, 20, 30]
list.remove(1);                        → [10, 20, 30]
for(int elem: list)
        SOP(elem + " ");               → 10 20 30.
```