```java
public class CharStack
{
        private final int MAX_CAP = 100;
        private int top; //location of top item on the stack
        private char [] s; //The stack

        public CharStack()
        //PRE:
        //POS:
        //TAS: Create an empty stack with a default capacity
        {
               s = new char[MAX_CAP];
               top = -1;
        }//CharStack Constructor

        public void push(char element)
        //PRE:
        //POS:
        //TAS: Add a new item to the stack
        {
          top ++;
          s[top] = element;
        }//push

        public char pop ()      //This method stubbed in
        //PRE:!isEmpty() && top > -1
        //POS:top(exit) == top<entry>-1
        //TAS: remove and return the item on the top of the stack
        {
           char ret = s[top];
           top --;
           return ret;
        }//pop

        public char peek ()
        //PRE:!isEmpty() && top > -1
        //POS:
        //TAS:Returns the top value of the stack.
        {
           return s[top];
        }//peek

       public String toString()
        //PRE:
        //POS:
        //TAS: Creates and returns all characters in the stack.
        {
           String ret = "";

           for(int i = 0; i < top; i++)
           {
               ret = ret + s[top];
           }//for

           return ret;

        }//toString

        public boolean isEmpty()    //This method is stubbed in
        //PRE:
        //POS:
        //TAS: Return whether or not the stack is empty. HINT: look at the constructor.
        {
          if(top == -1)
            return true;
          else
            return false;
        }//emptyStack

```

```
70          public boolean isFull()
71          //PRE:
72          //POS:
73          //TAS: return whether or not the stack is full
74          {
75            return top == MAX_CAP-1;
76          }//fullStack
77
78
79          public int size()
80          //PRE:
81          //POS:
82          //TAS: Returns the size of the stack.
83          {
84            return (top + 1);
85          }//size
86
87    }//CharStack
88
89
```