16/1/24

1. Demonstrate various string constructor with proper java program

```
class String {
    public static void main (String args[]){
            char c[] = {'J', 'a', 'v', 'a'};
            String s1 = new String (c);
            String s2 = new String(S1);
            System.out.println (s1);
            System.out.println(s2);
    }}
O/P : Java
      Java
```

8. Demonstrate startswith() to give output true or false.

```
public class Main {
    public static void main (String args[]) {
        String test = "testString";
        String pattern = "te";
        System.out.println(test.startswith(pattern));
        pattern = "est";
        System.out.println(test.startswith(pattern));
    }
}
O/P :
  True
  False
```

__/__/__

19. Write a Java program to create an abstract class Bird with abstract method fly() and makeSound(). Create subclass Eagle and Hawk that extend the Bird class and implement the respective methods to describe how each bird flies and makes a sound.

```
abstract class Bird {
        abstract void fly();
        abstract void makeSound();
    }

class Eagle extends Bird {
        void fly(){
            System.out.println("Eagle can fly very high ");
        }
        void makeSound(){
            System.out.println("Eagle makes a screech sound
                        ");

        }
    }

class Hawk extends Bird {
        void fly() {
            System.out.println("Hawk can fly moderatlly
                high");
        }
        void makeSound(){
            System.out.println("Hawk makes a shrill sound");
        }
    }
```

_/_/__

```
public class Main {
    public static void main(String args[]) {
        Bird bird1 = new Eagle();
        bird1.fly();
        bird1.makeSound();

        Bird bird2 = new Hawk();
        bird2.fly();
        bird2.makeSound();
    }
}
```

O/P :

```
Eagle can fly very high
Eagle makes a screech sound
Hawk can fly moderatlly high
Hawk makes a shrill sound
```

__/__/__

Write a Java program to create a generic class Stack which holds 5 integers and 5 double values.

```
import java.util.EmptyStackException;
public class Stack<T>{
  private int maxSize;
  private int top;
  private Object[] stackArray;

  public Stack(int size){
        maxSize = size;
        stackArray = new Object[maxSize];
        top = -1;
     }

  public void push(T value){
     if(top<maxSize-1){
            top++;
         StackArray[top] = value;
     }else{
      throw new RuntimeException
            ("Stack is full");
     }
   }

   @SupperssWarnings("unchecked")
    public T pop(){
     if(!isEmpty()){
        return (T)stackArray[top--];
}else {
    thrownew EmptyStackException();
  }
}

public boolean isEmpty(){
      return (top==-1);
  }

public int size() {
     return top+1;
}

public static void main(String args[]){
    Stack<Integer> intStack = new Stack<>(5);
    Stack<Double> doubleStack = new Stack<>(5);

  for (int i=0 ; i<5; i++) {
        intStack.push(i);
        doubleStack.push(i);
  }

  System.out.println("Integer Stack");
  for (int i=0; i<5; i++) {
      System.out.println(intstack.pop());
   }
  System.out.println("Double Stack");
  for (int i=0; i<5 ; i++) {
      System.out.println(doubleStack.pop()); }
 }
}
```

O/P:

```
Integer Stack:
 4
 3
 2
 1
 0
Double Stack:
 4.0
 3.0
 2.0
 1.0
 0.0
```

[illegible] 17/01/24