

Unit 11 - Week 9

Course outline

How does an NPTEL online course work?

Week 0

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Week 9

JACK Compiler: Lexical Analysis - Tokenization

Project 10: Compiler for JACK - Part-1 Demo

The JACK Grammar

Compiler for JACK: Parsing the JACK Program

Project 10: Part 2: The Token Analyzer

Project 10: Part 3: Testing the Correctness

Quiz : Assignment 9

Week 10

week 11

Week 12

Text Transcripts

Download Videos

Assignment 9

The due date for submitting this assignment has passed. As per our records you have not submitted this assignment.

Due on 2020-04-01, 23:59 IST.

1) How many lexical tokens does the following JACK code contain?

```
let x = "Hello World!";
```

No, the answer is incorrect. Score: 0

Accepted Answers: (Type: Numeric) 5

1 point

2) How many integer tokens are present in the given JACK program?

```
var int a;
var String b;
let a = 201;
let b = "204";
```

No, the answer is incorrect. Score: 0

Accepted Answers: (Type: Numeric) 1

1 point

3) Which of the following is(are) NOT valid tokens in JACK?

- `__x__`
- `2_x_x`
- `void_`
- `int_`

No, the answer is incorrect. Score: 0

Accepted Answers: `2_x_x`

1 point

4) Which of the following will occur when the given JACK code snippet is passed through the frontend of the JACK compiler?

```
let x = 1;
y = 2;
let z = 3;
```

- Lexical error
- Parse error
- Compile without any errors
- Variable declaration error

No, the answer is incorrect. Score: 0

Accepted Answers: `Parse error`

1 point

5) Which of the following error(s) can be thrown by the tokenizer?

- Syntax error: Expected ;
- Variable undeclared in the current scope
- Invalid token `1_`
- Invalid character `!`

No, the answer is incorrect. Score: 0

Accepted Answers: `Invalid token 1_`
`Invalid character '!'`

1 point

6) Say True or False: In the grammar for JACK language, the terminal element is always a lexical token.

- TRUE
- FALSE

No, the answer is incorrect. Score: 0

Accepted Answers: `TRUE`

1 point

7) Which of the following keyword(s) can be present as the first lexical element when a {tt statement} is expanded using recursive descent parsing?

- `let`
- `if`
- `while`
- `field`
- `static`

No, the answer is incorrect. Score: 0

Accepted Answers: `let`
`if`
`while`

1 point

8) Identify ALL the production rules which will be executed when the following JACK code is parsed using the recursive descent parsing

```
let x = 2 + y;
```

- `term`
- `unaryOp`
- `expression`
- `statement`
- `expressionList`
- `letStatement`

No, the answer is incorrect. Score: 0

Accepted Answers: `term`
`expression`
`statement`
`letStatement`

1 point

9) Identify all the CORRECT statement(s).

- The `.xml` file generated from JACK program can be used to translate to multiple VMs, if they exist.
- The same grammar used for JACK language can be used for other high-level languages like Java as well.
- Inside the body of a subroutine, all `VarDec` should be present before statements.
- Inside the body of a class, all `classVarDec` should be present before `subroutineDec`.

No, the answer is incorrect. Score: 0

Accepted Answers: `The .xml file generated from JACK program can be used to translate to multiple VMs, if they exist.`
`Inside the body of a subroutine, all VarDec should be present before statements.`
`Inside the body of a class, all classVarDec should be present before subroutineDec.`

1 point

10) Assume you already have a complete JACK compiler which can generate binary code for JACK programs. Now, we write a new compiler for JACK to HACK binary translation in JACK language itself. Is it possible for the existing JACK compiler to 'compile' the new compiler?

- Yes
- No
- Maybe

No, the answer is incorrect. Score: 0

Accepted Answers: `Yes`

1 point