Quiz 1: INTRO

Question 1 of 10 1.0 Points

```
int x[10];

x[10] = 10;
```

C++ lets you write to an element outside the array. Based on this design, which criteria of C++ is negative?

A. Readability

B. Reliability

C. Cost

D. Writability

Question 2 of 10 1.0 Points

In Algol. There is no reserved word, so there may be a valid if statement like: if if = then then. Based on this design, which criteria is negative?

A. Reliability

B. Cost

C. Readability

D. Writability

Question 3 of 10 1.0 Points

Although being statically type checking, Java performs dynamic type checking. For example, Java is able to prevent an access to an element outside an array such as

```
int a[10], y;

y = 12;

x[y] = 12;
```

which criteria is negative by this design?

A. Cost

B. Writability

C. Readability

D. Reliability

Question 4 of 10 1.0 Points

Given the following code that contains an syntax error at the second line:

cout << 120;

x = y +; //syntax error

If the value 120 is printed out before the error message is provided, which translator is used?

A. Compiler
B. Hybrid Compiler and Interpreter
C. Just-In-Time Compiler
D. Pure Interpreter

Question 5 of 10 1.0 Points

In the beginning, a program will be executed by an interpreter but after a certain time, a method in the program which is executed many times will be translates by ___________ into machine code.

A. Compiler
B. Hybrid Compiler and Interpreter
C. Just-In-Time Compiler
D. Pure Interpreter

Question 6 of 10 1.0 Points

Assume that:

- A program that contains n lines of code

- It takes t1 seconds to translate each line of code, It takes t2 seconds to execute machine code of each line of code. The program is translated and executed only one time

- No cost to switch between translation and execution which interpreter or compiler will make the program run faster?

A. Compiler
B. Compiler and Interpreter make the program run equally
C. Interpreter
D. Cannot determine which will make the program run faster

Question 7 of 10 1.0 Points

Given the following C++ code that contains the error in the first line:

int wrong@id;

Which phase of compilation process will issue the error message?

- **A. Lexical analyzer**
- B. Semantic analyzer
- C. Syntax analyzer
- D. Intermediate code generator

Question 8 of 10 1.0 Points

Given the following C++ code that contains a type mismatch error at line 2:

int x;

x = 10.2;

Which compilation phase will issue the type mismatch error message?

- A. Syntax analyzer
- **B. Semantic analyzer**
- C. Lexical analyzer
- D. Intermediate code generator

Question 9 of 10 1.0 Points

Which program performs file inclusion?

- **A. Preprocessor**
- B. Assembler
- C. Loader
- D. Linker

Question 10 of 10 1.0 Points

A program can be made of many source files but there is only one execution file of the program. Which of the following combines many object files into one execution file?

A. Assembler
B. Loader
C. Linker
D. Preprocessor

Quiz 2: OOP

Question 1 of 10 1.0 Points

Assume that there are 2 members, field1 and foo, declared in class A and there are also 2 members, fields2 and fie, declared in class B, and A is the superclass of B, how many members in class A and class B?

A. A and B has 2 members
B. A has 2 members while B has 4 members
C. A and B has 4 members
D. A has 4 members while B has 2 members

Question 2 of 10 1.0 Points

Assume that A is the superclass of B and C, and variables x, y, z are declared as follows

A x;

B y;

C z;

Select valid assignment statement(s)

A. x = new B;
B. x = new C;
C. y = new A;
D. z = new B;
E. z = new A;

Question 3 of 10 1.0 Points

Assume that class A is the superclass of class B and class B is the superclass of class C. There is method foo declared in class A and it is overridden in class B and also class C (i.e. foo is declared in B and C). Variables x, y, z are declared as follows

A x;

B y;

C z;

Match the call to the set of its targets.

A. {foo in C}
B. {foo in B, foo in C}
C. {foo in A, foo in B}
D. {foo in A, foo in B, foo in C}

x.foo() -> D

y.foo() -> B

z.foo() -> A

not match -> C

Question 4 of 10 1.0 Points

Assume that class A is the superclass of class B and class B is the superclass of class C. There are declarations of static method foo in class A, B and C.

A. A.foo() may call foo defined in A, in B or in C
B. B.foo() may call foo defined in A or in B
C. B.foo() may call foo defined in B or in C
D. A.foo() may call just foo defined in A
E. B.foo() may call just foo defined in B

Question 5 of 10 1.0 Points

To declared a static fields *sfields* for class ABC in Scala, how must programmers write?

A. Add the following line in class ABC, var sfields: Int
B. Add the following line in object ABC, static var sfields: Int
C. Add the following line in object ABC, var sfields: Int
D. Add the following line in class ABC, static var sfields: Int

Question 6 of 10 1.0 Points

In Scala, to declare class ABC as a subclass of class DEF, how must programmers write?

A. class DEF extends ABC
B. class DEF super ABC
C. class ABC extends DEF
D. class ABC: DEF

Question 7 of 10 1.0 Points

Given the following code in Scala,

abstract class A { def cal(x: Int): Int }

trait B extends A { abstract override def cal(x: Int): Int = super.cal(x * 2) }

trait C extends A { abstract override def cal(x: Int): Int = super.cal(x + 1) }

class D extends A { def cal(x: Int): Int = x }

val t = new D with B with C

println(t.cal(5))

What is the printed value?

A. 5
B. 11
C. 12
D. Other value

Question 8 of 10 1.0 Points

How to define a member which can only be accessed inside the object? For example, class X {

var f: Int // <= how to declare this field to be accessed just inside an object of X

var m = new X

//this.f is Ok but m.f is wrong in the code of class X

}

A. private var f: Int
B. private[this] var f: Int
C. protected var f: Int
D. var f: Int
E. protected[this] var f: Int

Question 9 of 10 1.0 Points

How to create an object of a case class in Scala? For example, for the following case class, case class Rational(n: Int, d: Int)

Please select the shortest correct answer.

A. new Rational(2, 3)
B. create Rational(2, 3)
C. object Rational(2, 3)
D. Rational(2, 3)

Question 10 of 10 1.0 Points

Which is/are the declaration(s) of A such that the new A is INVALID?

A. case class A
B. trait A
C. class A
D. abstract class A

Quiz 3: Functional Programming

Question 1 of 10 1.0 Points

Which is the foundation theory of functional programming?

A. Algebra
B. Von Neumann Architecture
C. Lambda Calculus
D. Predicate Logic

Question 2 of 10 1.0 Points

Which is NOT a property of functional programming language?

A. Loop structures controlled by conditional expression
B. Immutable variable
C. A function can return another function
D. A function can return function as a parameter

Question 3 of 10 1.0 Points

How many input parameter does an apply-to-all function require?

A. 1
B. 2
C. 3
D. 4

Question 4 of 10 1.0 Points

What is the result of the Insert Right function that receives:

- the first argument is function minus (-)
- the second argument is 4
- the third argument is <5, 6, 7, 8>

A. -20
B. -4
C. 2
D. Other value

Question 5 of 10 1.0 Points

A function is defined with n parameters, but it may be invoked with just (n – 1) arguments where it returns a function with 1 parameter. What kind of the function is?

A. Closure

B. Higher-order function

C. Generic function

D. Curried function

Question 6 of 10 1.0 Points

Given the following Scala code,

List(1, 2, 3).map(___________)

Select the right answer to fill the blank to return List(1, 4, 9)

A. n => n * n

B. List(1, 4 , 9)

C. n => n + n

D. n * n

Question 7 of 10 1.0 Points

Which is the implementation of the factorial function?

A. def fact(n: Int) = 1.to(n).foldLeft(0)(_ * _)

B. def fact(n: Int) = 0.to(n).foldLeft(1)(_ * _)

C. def fact(n: Int) = 1.to(n).foldLeft(0)(_ + _)

D. def fact(n:Int)= 1.to(n).fodLeft(1)(_ * _)

Question 8 of 10 1.0 Points

Select the right implementation(s) for the sum function? (There are some syntax error in the code to prevent copy and paste)

A. def sum(lst: List[Int]): Int = if (lst.isEmpty) 0 else lst.head + lst.tail

B. def sum(lst: List[Int]) = lst.foldLeft(0)(_ + _)

C. def sum(lst: List[Int]): Int = lst.head + sum(lst.tail)

D. def sum(lst: List[Int]): Int = if (lst.isEmpty) 0 else lst.head + sum(lst.tail)

Question 9 of 10 1.0 Points

Matching the blank line in the following code to the code to implement the function insert that inserts *n* into the ascending list *lst.*

```
def insert(n: Int, lst: List[Int]): List[Int] =
        if (lst.isEmpty)
                line1
        else
                if (n > lst.head)
                        line2
                else
                        line3
```

A. List(n)
B. n :: lst
C. Nil
D. lst.head :: insert(n, lst.tail)

line1 -> A

line2 -> D

line3 -> B

No line -> C

Question 10 of 10 1.0 Points

Fill in the following Scala code by matching to implement a merge function that merges two ascending list of integers.

```
def merge(lst1: List[Int], lst2: List[Int]): List[Int] =
        if (lst1.isEmpty)
                line1
        else
                if (lst1.head > lst2.head)
                        line2
                else
                        line3
```

A. lst2
B. lst1.head :: merge(lst1.tail, lst2)
C. lst2.head :: merge(lst1, lst2.tail)
D. Nil

line1 -> A
line2 -> C
line3 -> B
no line -> D

Quiz 4: Lexical Analysis

Question 1 of 10 1.0 Points

Which are the roles of lexical analysis?

A. Read the input stream of characters
B. Check if the order of tokens follows the grammar
C. Generate the sequence of tokens
D. Check if a variable is declared before it is used.

Question 2 of 10 1.0 Points

The lexical analysis is

A. after the syntax analysis
B. before the syntax analysis
C. a part of front end
D. a part of back end

Question 3 of 10 1.0 Points

How many tokens are there in the following C snippet?

if (a == b12 * 2) return a * b12;

A. 3
B. 10
C. 13
D. None of the above

Question 4 of 10 1.0 Points

How many tokens are there in the following C snippet?

x = -12*(a-1);

A. 8
B. 9
C. 10
D. 11

Question 5 of 10 1.0 Points

How many tokens are there in the following C snippet?

/* this is a block comment in C*/

x = “this is a string in C”; // this is a line c

A. 4
B. 6
C. 20
D. more than 20

Question 6 of 10 1.0 Points

List 5 shortest strings belonging to the language expressed by the following regular expression: (a|ab)*

A. Ɛ, a, aa, ab, aaa
B. Ɛ, a, ab, aab, aba
C. Ɛ, a, b, aa, ab
D. a, aa, ab, aba, aaa

Question 7 of 10 1.0 Points

List 5 shortest strings belonging to the language expressed by the following regular expression: ac*b

A. a, ab, acb, accb, acacb
B. ab, acb, accb, acccb, accccb
C. a, b, c, ab, cb
D. ab, acb, acacb, acacacb, acacacacb

Question 8 of 10 1.0 Points

List 5 shortest strings belonging to the language expressed by the following regular expression: ab?b?(a|b)

A. abb, abba, abbb, abbaa, abbba
B. aba, abb, abba, abbb, abab
C. aa, ab, aba, abb, abba
D. a, ab, aa, abb, abba

Question 9 of 10 1.0 Points

Write the regular expression to represent the language of any string that contains exactly two a's. For example, baba, bbbaa, abab.

A. b+ab+ab+
B. (a|b)*aa(a|b)*
C. b*ab*ab*
D. aa

Question 10 of 10 1.0 Points

Write the regular expression to represent the language of any string that contains only two strings bab and ab, i.e, {bab, ab}.

A. B(a|b)*
B. bab|ab
C. b?ab
D. (b|Ɛ)ab

Quiz 5: Lexical Analysis

Question 1 of 9 1.0 Points

A context-free grammar has (check all that apply)

A. at least a non-terminal symbol in the left hand side of any production
B. just only one non-terminal symbol in the left side of any production
C. unlimited number of terminal and non-terminal symbols in the right side of any production
D. at most one non-terminal symbol in the right hand side of any production

Question 2 of 9 1.0 Points

Given the following grammar

S => aSbS | c

Which of the following derivation processes are valid? Check all that apply

A. S ⇒ aSbS ⇒ acbc
B. S ⇒ aSbS ⇒ aaSbS ⇒ aacbS ⇒ aacbbS ⇒ aacbbc
C. S ⇒ aSbS ⇒ acbS ⇒ aacbS ⇒ aacbc
D. S ⇒ aSbS ⇒ acbS ⇒ acbc

Question 3 of 9 1.0 Points

Given the following grammar

S →A B

A → a∈A a |

B → b B | b

Which of the following derivation processes are valid? Check all that apply

A. S ⇒ AB ⇒ B ⇒ b
B. S ⇒ AB ⇒ AbB ⇒ Abb ⇒ aAabb ⇒ aaAaabb ⇒ aaaabb
C. S ⇒ AB ⇒ AbB ⇒ aAabB ⇒ aabb
D. S ⇒ AB ⇒ aAaB ⇒ aaAaaB ⇒ aaAaab ⇒ aaaab

Question 4 of 9 1.0 Points

Which of the following grammars are ambiguous? Check all that apply

A. S → S Sb | a |
B. A → A c A | b
C. S → a S | b S | a | b
D. A → A c B | B
B → y B z | t | x B

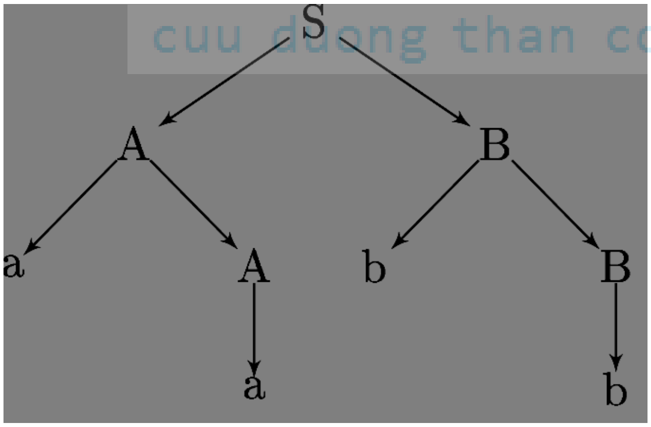
Question 5 of 9 1.0 Points

Based on the following grammar, which of the following parse trees are valid for input aabb? Check all that apply

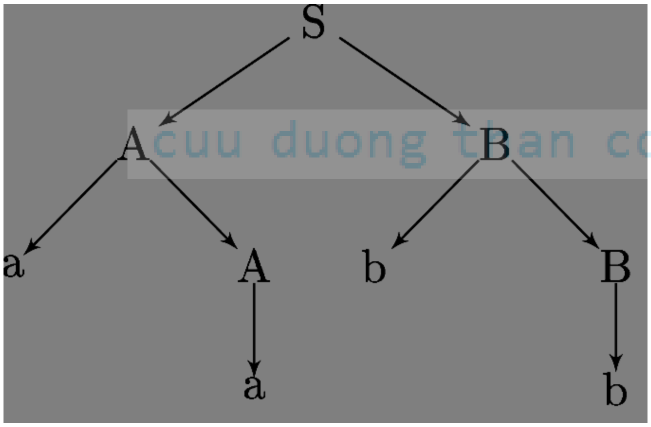
S → A B

A → a∈A a |

B → b B | b

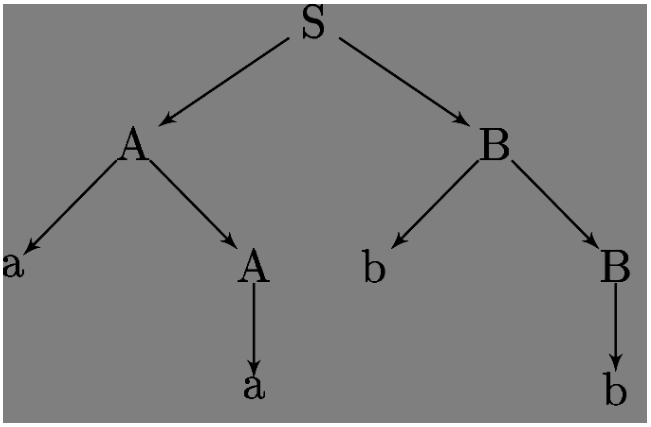


A.

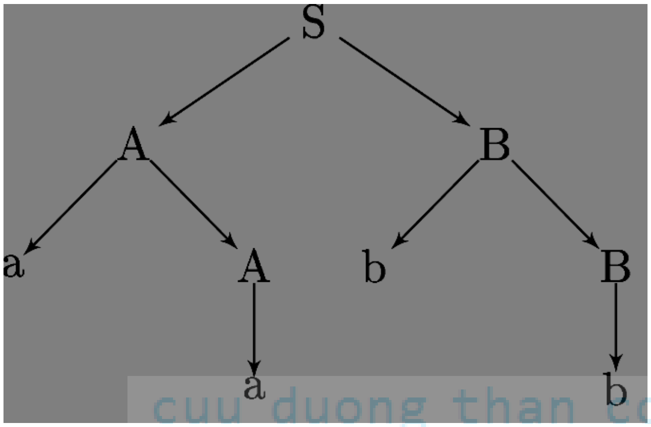


B.

C.



D.



Question 6 of 9 1.5 Points

Assume that an expression in a language is defined as follows

<exp> → <term> ASSIGN <exp> | <term>

<term> → <term> EXPONENT <fact> | <term> A

<fact> → <operand> RELOP <operand> | <oper

<operand> → LB <exp> RB | I

Which of the following sentences are correct? Check all that apply

A. ASSIGN is left-associated

B. ASSIGN is non-associated

C. ASSIGN is right-associated

D. EXPONENT is left-associated

E. EXPONENT is non-associated

F. EXPONENT is right-associated

G. ADDOP is left-associated

H. ADDOP is non-associated

I. ADDOP is right-associated
J. RELOP is left-associated
K. RELOP is non-associated
L. RELOP is right-associated

Question 7 of 9 1.5 Points

Assume that an expression in a language is defined as follows

<exp> → <term> ASSIGN <exp> | <term>

<term> → <term> EXPONENT <fact> | <term> A

<fact> → <operand> RELOP <operand> | <oper

<operand> → LB <exp> RB | ID

Which of the following sentences are correct? Check all that apply

A. ASSIGN has the highest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
B. ASSIGN has the lowest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
C. ASSIGN has equal precedence to another operator
D. EXPONENT has the highest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
E. EXPONENT has the lowest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
F. EXPONENT has equal precedence to another operator
G. ADDOP has the highest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
H. ADDOP has the lowest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
I. ADDOP has equal precedence to another operator
J. RELOP has the highest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
K. RELOP has the lowest precedence among operators ASSIGN, EXPONENT, ADDOP, RELOP
L. RELOP has equal precedence to another operator

Question 8 of 10 1.0 Points

Assume that an expression in a language is defined as follows

<exp> → <term> ASSIGN <exp> | <term>

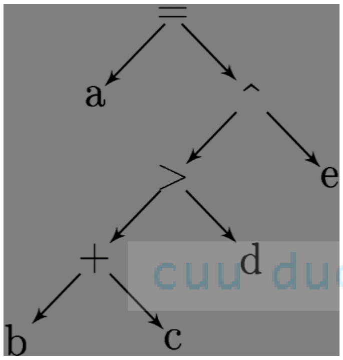
<term> → <term> EXPONENT <fact> | <term> A

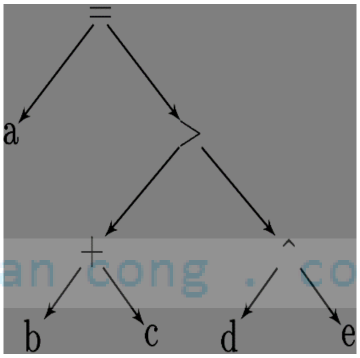
<fact> → <operand> RELOP <operand> | <oper

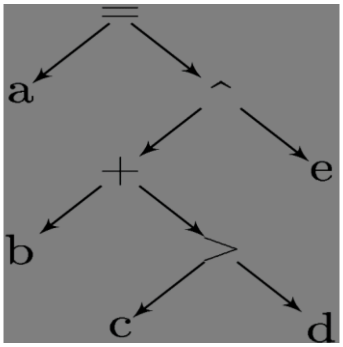
<operand> → LB <exp> RB | ID

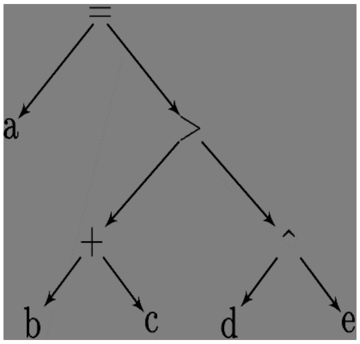
Let ID be token of identifiers, ASSIGN RELOP of
of ‘>’, LB of ‘(’ and RB of ‘)’.

Which of the following trees is the abstract syntax tree of expression a = b + c > d ^ e ?

A. 

C. 

B. 

D. 

Question 9 of 10 1.0 Points

Assume that an expression in a language is defined as follows

<exp> → <t eNr<exp> | <term> ASSIG

<term> → <term> EXPONENT <fact> | <term> A

<fact> → <operand> RELOP <operand> | <oper

<operand> → LB <exp> RB | ID

Let ID be token of identifiers, ASSIGN of
of ‘>’, LB of ‘(’ and RB of ‘)’.

Association and precedence help to reduce the brackets in an expression. For example, if operator - is left-associated, ((a - b) - c) can be rewritten as a - b - c with the same meaning. With the above grammar, which of the following expressions are equivalence to expression (a = ((b + c) > ((c = (a + b)) ^d)))? Check all that apply

A. a = b + c > c = a + b ^ d
B. a = b + c > (c = a + b) ^ d
C. a = b + c > ((c = a + b) ^ d)
D. a = (b + c) > (c = a + b ^ d)
E. (a = b + c) > (c = a + b ^ d)
F. a = (b + c) > ((c = a + b) ^ d)
G. a = ((b + c) > ((c = a + b) ^ d))
H. a = ((b + c) > (c = (a + b) ^ d))

Quiz 6: Parser

Question 1 of 10 1.0 Points

Given the following paragraph,

Each method declaration has the form:

<return type> <identifier>::<identifier> (<list of parameters>){<body>}

The <return type> is the type which the method will return. The return type of a method may be any type described in Section Type. The first <identifier> is the name of the class which the method belongs to. The second <identifier> is the name of the method. Each method prototype in the class declaration must have its corresponding method declaration. The <list of parameters> is a nullable semicolon-separated list of identifiers declared as the method's formal parameters.

Select the sentence(s) used to define the grammar?

A. Each method prototype in the class declaration must have its corresponding method declaration.

B. <return type> <identifier>::<identifier> (<list of parameters>){<body>}

C. The <return type> is the type which the method will return. The return type of a method may be any type described in Section Type.

D. The <list of parameters> is a nullable semicolon-separated list of identifiers declared as the method's formal parameters.

E. The first <identifier> is the name of the class which the method belongs to. The second <identifier> is the name of the method.

Question 2 of 10 1.0 Points

What is the equivalent EBNF of the following grammar?

<idlist> → ID COMMA <idlist> | ID

A. <idlist> → ID()* ID COMMA

B. <idlis t(COMMA → ID)+D

C. <idlist> → ID (COMMA ID)*

D. <idlist> → (ID COMMA)* ID

Question 3 of 10 1.0 Points

What is the equivalent EBNF of the following grammar?

stmtlist → (stmt)*

A. stmtlist → stmt stmtlist | stmt
B. stmtlist → stmtlist stmt | ∈
C. stmtlist → stmtlist stmt | stmt
D. stmtlist → stmt stmtlist | ∈

Question 4 of 10 1.0 Points

What is the equivalent EBNF of the following grammar? Check all that apply

<parlist> → <pardecls> | ε

<pardecls> → <pardecl> SEMICOLON <pardecls> | <pardecl>

A. <parlist> → (<pardecl> SEMICOLON <parde
B. <parlist> → (<pardecl>)? (SEMICOLON <pa
C. <parlist> → <pardecl>(SEMICOLON <pardec
D. <parlist> → (<pardecl>(SEMICOLON <parde

Question 5 of 10 1.0 Points

What is the BNF form of the following grammar in EBNF?

forstmt → "for" <id> = <exp> ("to" | "downto")

A. forstmt → "for" <id> = <exp> "to" <exp> ("by" <exp>)?"do" <stmt>
| "for" <id> = <exp> "downto" <exp> ("by" <exp>)?"do" <stmt>

B. forstmt → "for" <id> = <exp> ("to" | "downto") <exp> "do" <stmt>
| "for" <id> = <exp> ("to" | "downto") <exp> "by" <exp> "do" <stmt>

C. forstmt → "for" <id> = <exp> updown <exp> bystmt "do" <stmt>
updown → "to" | "downto"
bystmt → "by" <exp> | ∈

D. forstmt → "for" <id> = <exp> updown <exp> bystmt "do" <stmt>
updown → ("to" | "downto")
bystmt → ("by" <exp>)?

Question 6 of 10 1.0 Points

Given the following grammar,

```
S    →   B   |cC B |     D    f
B    →   ge    B    |
C    →   dc    C    |
D    h→| i
```

To derive from the nonterminal S on the input d, which production of S must be selected by a predictive parser?

A. S → B c

B. S → D f

C. S → C B

D. Cannot select any production because the grammar violates Condition 1

Question 7 of 10 1.0 Points

The following grammar does not satisfy Condition 1 to apply Predictive Parsing method. Which nonterminal symbol violates Condition 1?

```
S    →     BAc    b     |
A    →      CDB  f          |
B    →     eg     A     |
C    →   d∈    C     |
D    →  i    h     |
```

A. S

B. C

C. A

D. B

Question 8 of 10 1.0 Points

Given the following grammar,

```
S    →   | BAc   b
A    →      D    f          |          C     B
B    →      g     A     |          e
C    →   d∈   C     |
D    →      h               |          i
```

What is the First(Ab)?

A. {h, i, d}

B. {h, i, d, g, e}

C. {h, i, d, g, e, b}

D. {h, i, d, ∈ }

Question 9 of 10 1.0 Points

Given the following grammar,

E → T = E | T

T → T * F | F

F → (E) | id

Which one is the valid left recursion removal of the above grammar?

A.

E → E = T | T

T → F * T | F

F → (E) | i d

B.

E → T = E | T

T → F T '

T ' → *∈ F T ' |

F → (E) | i d

C.

E → T E '

E ' → =∈ T E ' |

T → F T '

T ' → *∈ F T ' |

F → (E) | i d

D.

E → E ' T

E ' E'⇒T | ∈

T → T ' F

T ' → *∈ F T ' |

F → (E) | i d

Question 10 of 10 1.0 Points

Given the grammar of stmt as follows,

stmt → IF expr THEN { stmt } ELSE { stmt }
| IF expr THEN { stmt }
| other
expr → TRUE | FALSE

which one left factors correctly the given stmt productions?

A.
stmt → stmt' | stmt' ELSE { stmt }
stmt' → IF expr THEN { stmt }
| other
expr → TRUE | FALSE

B.
stmt → IF expr stmt'
| other
stmt' → THEN { stmt } ELSE { stmt }
| THEN { stmt }
expr → TRUE | FALSE

C.
stmt → IF expr THEN { stmt } stmt'
| other
stmt' → ELSE { stmt } |
expr → TRUE | FALSE

D.
stmt → IF TRUE THEN { stmt } ELSE { stm
| IF FALSE THEN { stmt } ELSE { stmt }
| IF TRUE THEN { stmt }
| IF FALSE THEN { stmt }
| other

Quiz 7: AST

Question 1 of 10 1.0 Points

Given the following AST structure in Scala

trait Exp

case class BinExp(op:String,e1:Exp,e2:Exp) extends Exp

case class UnaExp(op:String,e:Exp) extends Exp

case class Lit(i:Integer) extends Exp

case class Id(i:String) extends Exp

Which is the valid AST of the following expression? (The association and precedence of operators as defined in BKOOL)

23 - (12 + 6) * 4 / 5

A. BinExp("/", BinExp("-", Lit(23), BinExp("*", BinExp("+", Lit(12), Lit(6)), Lit(4))))

B. BinExp("-", Lit(23), BinExp("/", BinExp("*", BinExp("+", Lit(12), Lit(6)), Lit(4)), Lit(5)))

C. BinExp("-", Lit(23), BinExp("*", BinExp("+", Lit(12), Lit(6)), BinExp("/", Lit(4), Lit(5))))

D. BinExp("-", Lit(23), BinExp("*", BinExp("+", Lit(12), Lit(6)), Lit(4)), BinExp("/", Lit(5)))

Question 2 of 10 1.0 Points

Assume that "- + ! 4" is the valid unary expression and the operators in unary expressions are right-association, i.e., the last operator "!" in the above expression is calculated first and then operator "+" and the first operator "-" is calculated last. What is the AST of the above expression?

A. UnaExp("-",UnaExp("+",UnaExp("!",Lit(4))))

B. UnaExp("-",UnaExp("!",UnaExp("+",Lit(4))))

C. UnaExp("!",UnaExp("+",UnaExp("-",Lit(4))))

D. UnaExp("+",UnaExp("!",UnaExp("-",Lit(4))))

Question 3 of 10 1.0 Points

As "- + ! 4" is a valid unary expression, the recognizer for a unary expression is given as follows

def unaryExp: Parser[Any] = rep("-"|"+"|"!") ~ fact

As concerned in the previous question, the operators in unary expressions are right-association. Modify the above rule to return an AST for a unary expression.

A. def unaryExp: Parser[Expr] = rep("-"|"+"|"!") ~ fact ^^ { case il ~ a => il.foldLeft(a)((b,x) => UnaExp(b,x))}

B. def unaryExp: Parser[Expr] = rep("-"|"+"|"!") ~ fact ^^ { case il ~ a => il.foldRight(a)((b,x) => UnaExp(b,x))}

C. def unaryExp: Parser[Expr] = rep("-"|"+"|"!") ~ fact ^^ { case a ~ il => il.foldRight(a)((b,x) => UnaExp(b,x))}

D. def unaryExp: Parser[Expr] = rep("-"|"+"|"!") ~ fact ^^ {case a ~ il => il.foldLeft(a)((b,x) => UnaExp(b,x))}

Question 4 of 10 1.0 Points

Extend the above AST structure as follows

trait Stmt

case class Assign(i:String,e:Exp) extends Stmt

case class IfThenElse(e:Exp,s1:Stmt,s2:Stmt) extends Stmt

case class IfThen(e:Exp,s:Stmt) extends Stmt

Which is the valid AST of the following statement?

if (a > 3) a := 4;

A. IfThenElse(BinExp(">", Id("a"), Lit(3)), Assign("a", Lit(4)))

B. Assign("a", Lit(4))

C. IfThen(BinExp(">", Id("a"), Lit(3)), Assign("a", Lit(4)))

D. BinExp(">", Id("a"), Lit(3))

Because the above statement is IfThen Statement

Question 5 of 10 1.0 Points

Which is the valid AST of the following statement?

if (x >= 5) then a = 5; else a = 7;

A. IfThenElse(BinExpr(">=", Id(x), Lit(5)), Assign(Id(a), Lit(5)), Assign(Id(a), Lit(7)))

B. [BinExpr(">=", Id(x), Lit(5)), Assign(Id(a), Lit(5)), Assign(Id(a), Lit(7))]

C. IfThen(BinExpr(">=", Id(x), Lit(5)), Assign(Id(a), Lit(5)), Assign(Id(a), Lit(7)))

D. IfThenElse("x >= 5", "a = 5", "a = 7")

Question 6 of 10 1.0 Points

Given the rule for recognizing an assignment statement as follows

def assignStmt: Parser[Any] = ident ~ "=" ~ exp ~ ";"

Modify this rule to make it return an AST for an assignment statement.

A. def assignStmt: Parser[Stmt] = ident <~ "=" ~> exp <~ ";" ^^ {case i ~ e => Assign(i,e)}

B. def assignStmt: Parser[Stmt] = (ident <~ "=") ~ (exp <~ ";") ^^ {case i ~ e => Assign(i,e)}

C. def assignStmt: Parser[Stmt] = ident ~ ("=" ~> exp <~ ";") ^^ {case i ~ e => Assign(i,e)}

D. def assignStmt: Parser[Stmt] = ident ~ exp ^^ {case i ~ e => Assign(i,e)}

Question 7 of 10 1.0 Points

Given the rule for recognizing an if statement as follows

```
def ifStmt:Parser[Any] = "if" ~ exp ~ "then" ~ stmt ~ opt("else" ~ stmt)
```

Modify it to make it return an AST for an if statement.

```
A. def ifStmt: Parser[Stmt] = ("if" ~> exp) ~ ("then" ~> stmt) ~ opt("else" ~>
   stmt) => {
           case a ~ b ~ c => match c {
                   case Some(d) => IfThenElse(a,b,d)
                   case None => IfThen(a,b)
           }
   }
```

```
B. def ifStmt: Parser[Stmt] = "if" ~ exp ~ "then" ~ stmt ~ opt("else" ~ stmt) =>
   {
           case a ~ b ~ c ~ d ~ e=> match e  {
                   case Some(f) => IfThenElse(a,b,f)
                   case None => IfThen(a,b)
           }
   }
```

```
C. def ifStmt:Parser[Stmt] = "if" ~> exp <~ "then" ~> stmt <~ opt("else" ~> stmt) =>
   {case a ~ b ~ c => IfThenElse(a,b,c)}
```

```
D. def ifStmt:Parser[Stmt] = ("if" ~> exp <~ "then") ~ stmt ~ opt("else" ~> stmt)
   => {
           case a ~ b ~ c => match c {
                   case Some(d) => IfThenElse(a,b,d)
                   case None => IfThen(a,b)
           }
   }
```

Question 8 of 10 1.0 Points

Given AST type of a declaration as follows:

```
trait Decl
case class VarDecl(id:String,typ:Type) extends Decl //id:name of ONE variable declared
case class ConstDecl(id:String,e:Exp) extends Decl

trait Type
object IntType extends Type
object FloatType extends Type
```

Match the input and the result AST

A. IntType

B. VarDecl("a", IntType)

C. ConstDecl("a", Literal(4))

D. FloatType

E. Not match

1. **int a; B**
2. **a = 4; C**
3. **int A**
4. **float D**
5. **int a,b,c; E**

Question 9 of 10 1.0 Points

Given AST type of a declaration as follows:

```
trait Decl

case class VarDecl(id:String, typ:Type) extends Decl //id:name of ONE variable declared

case class ConstDecl(id:String, e:Exp) extends Decl

trait Type

object IntType extends Type

object FloatType extends Type
```

What is the result of the variable declaration: int a,b,c; ?

A. VarDecl("a b c", IntType)

B. List(VarDecl("a", IntType), VarDecl("b", IntType), VarDecl("c", IntType))

C. VarDecl("a", VarDecl("b", VarDecl("c", IntType)))

D. VarDecl(List("a","b","c"), IntType)

Question 10 of 10 1.0 Points

Given AST type of a declaration as follows:

trait Decl

case class VarDecl(id:String,typ:Type) extends Decl //id:name of ONE variable declared

case class ConstDecl(id:String,e:Exp) extends Decl

trait Type

object IntType extends Type

object FloatType extends Type

and given the grammar of a variable declaration as follows:

def vardecl:Parser[Any] = type ~ rep1sep(ident,",")

Based on the above grammar and the above AST, how to generate the result of a variable declaration?

A. def vardecl: Parser[Decl] = type ~ rep1sep(ident, ",") ^^ {case t ~ il => il.tail.foldLeft(VarDecl(il.head,t))((a,b) => VarDecl(a,b))}

B. def vardecl: Parser[List[Decl]] = type ~ rep1sep(ident, ",") ^^ {case t ~ il => il.map(VarDecl(_,t))}

C. def vardecl: Parser[Decl] = type ~ rep1sep(ident, ",") ^^ {case t ~ il =>VarDecl(il,t)}

D. def vardecl: Parser[Decl] = type ~ rep1sep(ident,",") ^^ {case t ~ il => VarDecl(il.foldLeft("")((a,b) => a + b),t)}

Quiz 8: Name

Question 1 of 10 1.0 Points

Given the following C code,

```
int p[10];
int foo(int x) {
    static int q [10] ;
    int r [10] ;
    int s = new int [10] ;
    switch ( x ) {
    case 1 : return p ;
    case 2 : return q ;
    case 3 : return r ;
    case 4 : return s ;
    default : foo ( x+1);
   }
}
```

Which following statements are WRONG? Check all that apply

A. Array r is allocated in stack memory

B. Array q is allocated in stack memory

C. Array p is allocated in stack memory

D. Array p is allocated in static memory

E. Array r is allocated in heap memory

F. Array pointed by s is allocated in heap memory

G. Array q is allocated in static memory

H. Array pointed by s is allocated in stack memory

Question 2 of 10 1.0 Points

Given the code in question 1, which following statements are CORRECT? Check all that apply

A. The lifetime of array p is equal to the lifetime of foo

B. The lifetime of array p is equal to the lifetime of whole program

C. The lifetime of array q is equal to the lifetime of foo

D. The lifetime of array p is equal to the lifetime of whole program

E. The lifetime of array r is equal to the lifetime of foo

F. The lifetime of array r is equal to the lifetime of whole program

G. The lifetime of array pointed by s is equal to the lifetime of foo

H. The lifetime of array pointed by s is equal to the lifetime of whole program

Question 3 of 10 1.0 Points

Given the code in question 1, which statements may cause runtime problem (dangling reference)?

A. return r;

B. return s;

C. return q;

D. return p;

Question 4 of 10 1.0 Points

Given the code in question 1, when foo is called recursively, which arrays are the same among foo executions? Check all that apply

A. p

B. q

C. r

D. pointed by s

Question 5 of 10 1.0 Points

Given the code in question 1, which array may become garbage? Check all that apply

A. pointed by s

B. q

C. r

D. p

Question 6 of 10 1.0 Points

What will happen when executing the following code?

```
int *p = new int;
int *q = p;
*p = 1;
```

A. dangling reference

B. polymorphism

C. garbage

D. alias

E. undeclared pointer

Question 7 of 10 1.0 Points

What is the binding time of the real address of variable count in the following C code?

```
static int count;
```

A. programming

B. compiling

C. running

D. loading

E. implementation

Question 8 of 10 1.0 Points

Given the following program

```
var x; //1
procedure sub1() {
   var x;//2
   call sub2();
}
procedure sub2() {
   x // use x
}
main(){
   call sub1();
}
```

Assume that the program is written in a static-scoping language and calling chain is main to sub1 to sub2, which declaration is referred by x in sub2?

A. it is referred to x that is declared at //1 in one time and at //2 in another time

B. it is referred to x that is declared at //2

C. it is referred to x that is declared at //1

D. error message: x is undeclared

Question 9 of 10 1.0 Points

Given the code in question 8, assume that the program is written in a static-scoping language, select all statements that are CORRECT

A. The referencing environment of x declared at //2 is sub2 and main

B. Scope of main is x declared at //1, sub1 and sub2

C. The referencing environment of main is x declared at //1, sub1 and sub2.

D. Scope of x declared at //1 is sub2 and main

Question 10 of 10 1.0 Points

Given the code in question 8, if the program is written in a dynamic-scoping language and calling chain is main to sub1 to sub2, which declaration of x is referred by x in sub2?

A. it is referred to x that is declared at //1

B. it is referred to x that is declared at //2

C. it is referred to x that is declared at //1 in one time and at //2 in another time

D. error message: x is undeclared

Quiz 9: Data Type

Question 1 of 10 1.0 Points

Which statement is NOT true? (Character)

A. C and C++ use ASCII code.

B. Java uses UNICODE code.

C. A value in character type is stored as a number representing the ASCII/UNICODE code of the value.

D. UNICODE code is originally used in C#.

Question 2 of 10 1.0 Points

Which statement is NOT true? (Enumerate)

A. In C#, enumerate type variable can't be assigned a value outside its defined range.

B. Enumerate types make program more readability.

C. In C#, enumerate type variables are coerced into integer types.

D. Enumerate types are implemented as integers.

Question 3 of 10 1.0 Points

Which statement is NOT true? (Float)

A. 011111000 equals to 28 (using IEEE 754 standard with exponent part has 3 bits)

B. In IEEE 754 standard, floating point data type needs at least 8 bytes.

C. Floating point data type can't model real number exactly.

D. In floating point data type, the number of bits use for fraction part is larger than the number of bits use for exponent part.

Question 4 of 10 1.0 Points

Which statement is NOT true? (Boolean)

- **A. To implement Boolean data type, we use at least 8 bits.**
- B. Boolean data type is the simplest one.
- C. Boolean data type has only two values.
- D. Using Boolean data type makes program more readability.

Question 5 of 10 1.0 Points

Which statement is NOT true? (Integer)

- **A. Integer data type is the simplest data type.**
- B. C and C++ offer unsigned integers.
- C. Java doesn't offer unsigned integers.
- D. Integer data type is supported directly by hardware.

Question 6 of 10 1.0 Points

Which statement is NOT true? (Union)

- **A. The size of a union variable is identified at run time.**
- B. C and C++ don't support type checking in union type.
- C. Union type is an unsafe construct.
- D. Java and C# don't have union type.

Question 7 of 10 1.0 Points

Which statement is NOT true? (Array)

- A. Storage allocation of stack-dynamic array is done at run time.
- B. Heap-dynamic array size can change at run time.
- C. Fixed stack-dynamic array does not run efficiently.
- **D. Static arrays are efficient and flexible.**

Question 8 of 10 1.0 Points

Which statement is NOT true? (Record)

A. Elements in record may have different types.

B. Record use dynamic subscripting.

C. In C++, objects are used as records.

D. In C++, there are two main types of record member reference.

Question 9 of 10 1.0 Points

Which statement is NOT true? (Decimal)

A. Java offers decimal data type.

B. COBOL is used essentially in business application.

C. C# offers decimal data type.

D. Decimal data type uses effectively memory.

Question 10 of 10 1.0 Points

Which statement is NOT true? (String)

A. In C, C++, string is implemented as character array.

B. COBOL is used essentially in business application.

C. Java string length is identified at compile time.

D. C and C++ store the string length in the run time descriptor.

Quiz 10: Sequence Control

Question 1 of 10 1.0 Points

Rewrite the following infix expression using Cambridge Polish prefix format?

a + b * c + d + e * f

A. + + + a * b c d * e f

B. + a * b c d * e f

C. (+ (+ (+ a (* b c)) d) (* e f))

D. (((a + (b * c)) + d) + (e * f))

E. (+ a (* b c) d (* e f))

Question 2 of 10 1.0 Points

Rewrite the following infix expression using the Polish postfix format? Assume that all operators have 2 operands

a + b * c + d + e * f

A. a b c * d e f * +

B. a + b c * d + e f * +

C. a b + c * d + e f * +

D. a b c * + d + e f * +

E. a b c * d e f * + + +

Question 3 of 10 1.0 Points

Rewrite the following infix expression using the Polish prefix format? Assume that every operator has only 2 operands.

a + b * c + d + e * f

A. + a * b c d * e f

B. + + a * b c + d * e f

C. + a * b c + + d * e f

D. + + + a * b c d * e f

E. + a * b c + d * e f

Question 4 of 10 1.0 Points

Select all possible result of the following C expression:

a + (a = 2) * a

Assume that variable *a* contains value 3 before the above expression is evaluated.

A. 6

B. 8

C. 9

D. 10

E. 7

Question 5 of 10 1.0 Points

In some language like ADA, there are two kinds of AND/OR operator: short-circuit (AND THEN) and non-short circuit(AND).Select logical expressions that are required short-circuit evaluation?

A. (a > 0) && (b > 0)

B. (a != b) && (a * b > 0)

C. (a > 0) && (sqrt(a) > 2) //sqrt is the function to calculate the square root of its parameter

D. (i < a.length) && (a[i] > 0)

E. (a == 0) || ((b / a) > 5)

Question 6 of 10 1.0 Points

Given the grammar of if statement, select the language that there is NO dangling else problem

A. <stmt> → if <expr> then <stmt> else <st
| if <expr> then <stmt>

B. <stmt> → if (<expr>) <stmt> else <stmt>
| if (<expr>) <stmt>

C. <stmt> → if <expr> then <stmt> else <st
| if <expr> then <stmt> endif

D. <stmt> → if <expr> then <stmt> else <stmt>

E. <stmt> → if <expr> \n (<indentation> <stmt> \n)+ else \n (<indentation> <stmt> \n)+
| if <expr> \n (<indentation> <stmt> \n)+

Question 7 of 10 1.0 Points

which switch statement can be implemented using jump table?

A. switch (x) {

   1: command1; break;

   20: command2; break;

   37: command3; break;

   }

B. switch (x) {

   1: command1; break;

   2: command2; break;

   3: command3; break;

   }

C. switch (x) {

   1..2: command1; break;

   4   : command2; break;

   5..6: command3; break;

   }

D. switch (x) {

   1..20: command1; break;

   21 : command2; break;

   22..100: command3; break;

   }

Question 8 of 10 1.0 Points

Given the semantics of the for statement:

```
for i = <first> to <last> by <step> do <stmt>
```

as follows,

```
i = <first>
loop:
if i > <last> goto out
<stmt>
i = i + <step>
goto loop
out:
```

How many times does the following for statement iterate?

```
n := 10;
j = 1;
for i:= 1 to n by j do j++;
```

A. 10

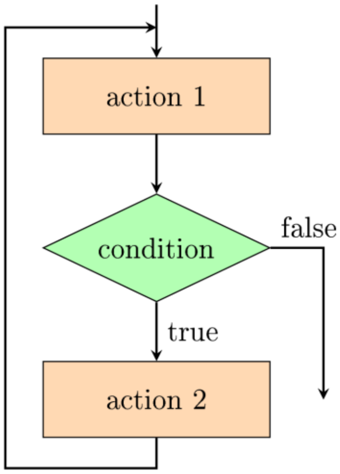
B. infinite

C. 15

D. 4

Question 9 of 10 1.0 Points

Given the following control flow,



Select the code that implements the above control flow?

A. while (condition) {

action 1

action 2

}

B. action 1

while (condition) {

action 2

action 1

}

C. for (action 1;condition;action 1) action 2

D. do {

action 1

if (condition) break

action 2

} while (true);

E. while (true){

```
action 1
if (! condition) break;
 action 2
};
```

Question 10 of 10 1.0 Points

Given the following C code,

```
while (i > n) { // line 1
    while (j > m) { //line 2
        while (k > p) {  // line 3
                s += k;
                if (s > 1000) break; // line 5
                k--;
        }
        j--; //line8
    }
    i--; //line 10
}
//line 12
```

Where does the control come when the break at line 5 is executed?

A. Line 3

B. Line 8

C. Line 12

D. Line 2

E. Line 10