1. (10 points) Java Language

1.1.

positive and negative infinities, and special Not-a-Number values

1.2.

Boolean, byte, short, int, long, char, float and double

1.3.

The initial value of int is 0 even if we do not assign any number to it.

1.4

A narrowing primitive conversion may lose information about the overall magnitude of a numeric value and may also lose precision and range.

It show warning message. It will be “possible lossy conversion from int to short”. However, if we forcefully cast like (short) int variable, there will be no warning message.

1.5

Packages and compilation unit may be stored in a distributed file system or some form of database.

If a host system stores packages and compilation units in a database, then the database must not impose the optional restrictions on compilation units permissible in file-based implementations.

1.6

An inner class is a nested class that is not explicitly or implicitly declared static.

1.7

The both Exception and Error classes are direct subclasses of Throwabl.

The class RuntimeException is a direct subclass of Exception. RuntimeException is the superclass of all the exceptions which may be thrown for many reasons during expression evaluation, but from which recovery may still be possible.

2. (15 points) Grammars and Regular Expressions

2.1

S -> S0 | S0S1

S0 -> aS0 | S0b | b

S1 -> cS1 | S1b | epsilon

2.2

S -> S0C | S1C | AS2 | AS3

S0 -> aS0b | aS0 | a

S1 -> aS1b | S1b | b

C -> cC | epsilon

S2 -> bS0c | bS0 | b

S3 -> bS3c | S3c | c

A -> aA | epsilon

2.3

(([a-zA-Z+].?)|([a-zA-Z0-9+](-?(([a-zA-Z0-9+]+-?))\*[a-zA-Z0-9+])).?)\*[^-]@((([a-zA-Z+].?)|([a-zA-Z0-9+](-?(([a-zA-Z0-9+]+-?))\*[a-zA-Z0-9+])).?)\*[^-]\.[a-zA-Z]{3}|(([12]?[0-9]?[0-9].){3}[0-9]))

If the length of input

5.

5.1

2+9-5-5 = 2+9-10 = 2-1 = -1

5.2

3+2\*12%2+3 = 3 + 2 \* 12 % 5 = 5\*12%5 = 60%5 =

5.3

8+6/2\*3+4 = 8+6/2\*7 = 14/2\*7 = 7\*7 = 49

5.4

6\*5/4+5-4\*3 = 6\*5/4+1\*3 = 6\*5/5\*3 = 30/5\*3 = 6\*3 = 18

6.

6.1

f ( f() && g() || h() ) {

cout << "What lovely weather!" << endl;

return true false; }

bool f() {

cout << "Hello, ";

return true;}

bool g() {

cout << "Darling. " << endl;

return false;}

bool h() {

cout << "World! " << endl;

return true;}

6.2

The C++ already has short circuit evaluation. At page 128 129 in C++ standard, Unlike &, && guarantees left-to-right evaluation: the second operand is not evaluated if the first operand is false. Unlike |, || guarantees left-to-right evaluation; moreover, the second operand is not evaluated if the first operand evaluates to true.

6.3

The Java compiler already has short circuit evaluation which is explained in Java Document at page 577 . The conditional-and operator && is like & (§15.22.2), but evaluates its right-hand operand only if the value of its left-hand operand is true.

The conditional-or operator || operator is like | (§15.22.2), but evaluates its righthand

operand only if the value of its left-hand operand is false.

7.

|  |  |  |
| --- | --- | --- |
| Unit | Var | Where to Declared |
| sub1 | x, y | sub1 |
| a, b | main |
| sub2 | x, t | sub2 |
| a, b | main |
| sub3 | y | sub3 |
| x, t | sub2 |
| a, b | main |