**Conyo++ Programming Language**

CSC617M

Chua, Jedrick L.

Livelo, Evan Dennison S.

Yao, John Paul S.

**Definition of Terms**

Conyogram - A Conyo++ program.

OMG - A prefix for declaring a constant.

Arte - A prefix for declaring a variable.

Yaya - A function in the program.

Like - A prefix for creating loops. < while loops? or for loops?

Utos - Program code blocks or statements.

Sabi-sabi - Expressions in the program.

Super Yaya - The main function in the program.

Sub Yaya - A user-defined class

MRW - My Reaction When

MDR - My Default Reaction

**Grammar Definition**

Conyo++ Grammar ‘G’ is defined by G = (V, T, P, S), where:

V = {}

T = {a, b, c, ..., z, A, B, C, ..., Z, 0, 1, 2, ..., 9, \_, :, ‘, “, +, -, \*, /, %, (, ), =, !, |, &, >, <, $, [, ], {, }, (all ascii characters)}

P = Productions are shown below.

S = <conyogram>

**PRODUCTIONS**

**Program**

<conyogram> **->** <OMG\_section> <yaya\_section> <sub\_yaya\_section>

<super\_yaya>

**Program Breakdown**

<OMG\_section> **->** <OMG\_list>

**->** ε

<OMG\_list> **->** <OMG\_dec>

**->** <OMG\_dec> <OMG\_list>

<OMG\_dec> **-> OMG** <IDENTIFIER> **makeKuha** <sabi\_sabi> **db**

<yaya\_section> **->** <yaya\_list>

**->** ε

<yaya\_list> **->** <yaya>

**->** <yaya> <yaya\_list>

<sub\_yaya\_section> **->** <sub\_yaya\_list>

**->** ε

<sub\_yaya\_list> **->** <sub\_yaya>

**->** <sub\_yaya> <sub\_yaya\_list>

<super\_yaya> **-> superYaya** <utos\_block>

**Literals, Data Types, Names**

<literal> **->** <INT\_LITERAL>

**->** <FLOAT\_LITERAL>

**->** <STRING\_LITERAL>

**->** <CHAR\_LITERAL>

**->** <BOOL\_LITERAL>

**->** <WALEY>

<data\_type> **->** <primitive\_dt>

**->** <reference\_dt>

<primitive\_dt> **-> inty**

**-> floaty**

**-> stringy**

**-> chary**

**-> booly**

<reference\_dt> **->** <valid\_name>

**->** <array\_dt>

<array\_dt> **->** <primitive\_dt> <brackets>

**->** <valid\_name> <brackets>

<valid\_name> **->** <IDENTIFIER>

**->** <valid\_name>**:**<IDENTIFIER>

<valid\_var> **->** <valid\_name>

**->** <valid\_name> <ref\_brackets>

**Class Declaration Productions**

<sub\_yaya> **-> makeUtusan** <IDENTIFIER> <sy\_body>

<sy\_body> **-> $** <sy\_opt> **$**

<sy\_opt> **->** <sy\_cont\_list>

**->** ε

<sy\_cont\_list> **->** <sy\_cont> <sy\_cont\_list>

**->** <sy\_cont>

<sy\_cont> **->** <arte\_dec>

**->** <yaya>

**->** <constructor>

<constructor> **-> makeSimula** <IDENTIFIER> <utos\_block>

<sy\_init> **-> brandNew** <reference\_dt> **(** <arte\_init\_list> **)**

**Variable Declaration Productions**

<arte\_dec> **-> makeArte** <data\_type> <IDENTIFIER> <arte\_assign> **db**

<arte\_assign> **-> makeKuha** <arte\_init>

<arte\_init> **->** <sabi\_sabi>

**->** <array\_init>

**->** <sy\_init>

**Function Declaration Productions**

<yaya> **->** <yaya\_header> <utos\_block>

<yaya\_header> **-> hireYaya** <IDENTIFIER> **bayad (** <yaya\_param\_sec> **)**

**makeBalik** <data\_type> <IDENTIFIER>

**-> hireYaya** <IDENTIFIER> **bayad (** <yaya\_param\_sec> **)**

**makeBalik poor**

<yaya\_param\_sec> **->** <yaya\_param\_list>

**->** ε

<yaya\_param\_list> **->** <yaya\_param>**,** <yaya\_param\_list>

**->** <yaya\_param>

<yaya\_param> **->** <data\_type> <IDENTIFIER>

**Array Productions**

<array\_init> **-> {** <arte\_init\_list> **}**

**-> { }**

<arte\_init\_list> **->** <arte\_init>**,** <arte\_init\_list>

**->** <arte\_init>

<brackets> **-> []** <brackets>

**-> []**

<ref\_brackets> **-> [** <sabi\_sabi> **]**

**Program Code Productions**

<utos\_block> **-> $** <utos\_block\_opt> **$**

<utos\_block\_opt> **->** <block\_content>

**->** ε

<block\_content> **->** <utos\_dec> <block\_content>

**->** <utos\_dec>

<utos\_dec> **->** <utos\_dec\_db> **db** **->** <utos\_dec\_val> **db**

**->** <utos\_dec\_nodb>

<utos\_dec\_nodb> **->** <utos\_likeKapag>

**->** <utos\_makePalit>

**->** <utos\_likeHabang>

**->** <utos\_makeGawaHabang>

**->** <utos\_makeUlit>

<utos\_dec\_db> **->** <utos\_makeKuha>

**->** <utos\_makeSulat>

**-> breakup**

**-> makeup**

<utos\_dec\_val> **->** <utos\_makeTawag>

**->** <utos\_makeBasa>

<utos\_makeKuha> **->** <valid\_var> <arte\_assign>

**->** <sabi\_sabi>

<utos\_makeTawag> **-> makeTawag** <valid\_name> **(** <arte\_init\_list> **)**

<utos\_makeSulat> **-> makeSulat (** <sabi\_sabi> **)**

<utos\_makeBasa> **-> makeBasa (** <data\_type> **)**

<utos\_likeKapag> **-> likeKapag (** <sabi\_sabi> **)** <utos\_block>

<utos\_thisNalang>

<utos\_thisNalang> **-> ε**

**-> thisNalang** <utos\_block>

**->** **thisNalangKapag (** <sabi\_sabi> **)** <utos\_block> <utos\_thisNalang>

<utos\_makePalit> **-> makePalit(** <sabi\_sabi> **) $** <makePalit\_MRW> **$**

<makePalit\_MRW> **-> MRW** <sabi\_sabi> **:** <utos\_block\_opt> <makePalit\_MRW>

**-> MDR :** <utos\_block\_opt>

**->** **ε**

<utos\_likeHabang> **->** **likeHabang (** <sabi\_sabi> **)** <utos\_block>

**-> likeHabang (** <sabi\_sabi> **) db**

<utos\_makeGawaHabang> **-> makeGawa** <utos\_block> **likeHabang**

**(**<sabi\_sabi>**)** **db**

<utos\_makeUlit> **-> makeUlit (** <utos\_makeKuha> **db** <sabi\_sabi>   
 **db** <utos\_makeKuha> **)** <utos\_block>

**-> makeUlit (** <valid\_var> **db** <sabi\_sabi>   
 **db** <utos\_makeKuha> **)** <utos\_block>

**Expression Productions**

<sabi\_sabi\_end> **->** <valid\_var>

**->** <utos\_makeTawag>

**->** <utos\_makeBasa>

**->** <literal>

<sabi\_sabi> **->** <ss\_OR>

<ss\_OR> **->** <ss\_AND> **||** <ss\_OR>

**->** <ss\_AND>

<ss\_AND> **->** <ss\_equality> **&&** <ss\_AND>

**->** <ss\_equality>

<ss\_equality> **->** <ss\_comparison> <equality> <ss\_equality>

**->** <ss\_comparison>

<ss\_comparison> **->** <ss\_a1> <comparison> <ss\_comparison>

**->** <ss\_a1>

<ss\_a1> **->** <ss\_a2> <arithmetic\_1> <ss\_a1>

**->** <ss\_a2>

<ss\_a2> **->** <ss\_unary\_1> <arithmetic\_2> <ss\_a2>

**->** <ss\_unary\_1>

<ss\_unary\_1> **-> !**<ss\_unary\_1>

**-> ++**<ss\_unary\_1>

**-> --**<ss\_unary\_1>

**->** <ss\_unary\_2>

<ss\_unary\_2> **->** <ss\_unary\_2>**++**

**->** <ss\_unary\_2>**--**

**->** <ss\_paren>

<ss\_paren> **-> (**<ss\_paren>**)**

**->** <sabi\_sabi\_end>

<equality> **-> ==**

**-> <>**

<comparison> **-> >=**

**-> <=**

**-> >**

**-> <**

<arithmetic\_1> **-> +**

**-> -**

<arithmetic\_2> **-> \***

**-> /**

**-> %**

<sabi\_sabi\_end> **->** <valid\_var>

**->** <utos\_makeTawag>

**->** <utos\_makeBasa>

**->** <literal>

<sabi\_sabi> **->** <number\_sabi\_sabi>

**->** <boolean\_sabi\_sabi>

**->** <string\_sabi\_sabi>

<number\_sabi\_sabi> **->** <number\_level\_1> **+** <number\_sabi\_sabi>

**->** <number\_level\_1> **-** <number\_sabi\_sabi>

**->** <number\_level\_1>

**->** <number\_increment>

<number\_level\_1> **->** <number\_level\_2> **\*** <number\_level\_1>

**->** <number\_level\_2> **/** <number\_level\_1>

**->** <number\_level\_2> **%** <number\_level\_1>

**->** <number\_level\_2>

<number\_level\_2> **-> (** <number\_level\_2> **)**

**->** <sabi\_sabi\_end>

<number\_increment> **->** <number\_level\_2> **++**

**->** <number\_level\_2> **--**

<boolean\_sabi\_sabi> **->** <number\_sabi\_sabi> <compare> <number\_sabi\_sabi>

**->** <string\_sabi\_sabi> **==** <string\_sabi\_sabi>

**->** <string\_sabi\_sabi> **<>** <string\_sabi\_sabi>

**->** <boolean\_level\_1> **==** <boolean\_sabi\_sabi>

**->** <boolean\_level\_1> **<>** <boolean\_sabi\_sabi>

**->** <boolean\_level\_1>

<boolean\_level\_1> **->** <boolean\_level\_2> **||** <boolean\_level\_1>

**->** <boolean\_level\_2>

<boolean\_level\_2> **->** <boolean\_level\_3> **&&** <boolean\_level\_2>

**->** <boolean\_level\_3>

<boolean\_level\_3> **-> (** <boolean\_level\_3> **)**

**-> !**<boolean\_level\_3>

**->** <sabi\_sabi\_end>

<compare>  **-> ==**

**-> >=**

**-> <=**

**-> >**

**-> <**

**-> <>**

<string\_sabi\_sabi> **->** <sabi\_sabi\_end> **+** <string\_sabi\_sabi>

**->** <sabi\_sabi\_end>

**Miscellaneous Productions**

<INT\_LITERAL> **->** <SIGN> <NUMBERS+>

<FLOAT\_LITERAL> **->** <INTEGER>**.**<NUMBERS+>

**->** <SIGN>**.**<NUMBERS+>

**->** <INTEGER>**.**<NUMBERS+>**e**<INTEGER>

**->** <INTEGER>**e**<INTEGER>**.**<NUMBERS+>**e**<INTEGER>

<CHAR\_LITERAL> **-> ‘**<ASCII>**’**

<STRING\_LITERAL> **-> “**<ASCII\*>**”**

<BOOL\_LITERAL> **-> yuhh**

**-> nuhh**

<WALEY> **-> waley**

<NUMBERS+> **->** <NUMBER>

**->** <NUMBER> <NUMBERS+>

<NUMBER> **-> 1**|**2**|**3**|**4**|**5**|**6**|**7**|**8**|**9**|**0**

<LETTER> **-> a**|**b**|**c**|..|**z**|**A**|**B**|**C**|..|**Z**

<ASCII\*> **->** <ASCII> <ASCII\*>

**->** ε

<ASCII> **->** *any ascii character*

<IDENTIFIER> **->** <LETTER> <ID\_CONT>

**-> \_**<ID\_CONT>

<ID\_CONT> **->** <LETTER> <ID\_CONT>

**-> \_**<ID\_CONT>

**->** <NUMBER> <ID\_CONT>

**->** ε