FACULTY OF ENGINEERING

CHULALONGKORN UNIVERSITY

2110316 PROG LANG PRIN

Year 3rd, First Semester, Final Examination December 12, 2018. Time 8:30-11:30

ชื่อ-นามสกุล....................................................………....เลขประจำตัว...…….……....……......ตอนเรียนที่…….….เลขที่ในCR58.……...

หมายเหตุ

1. ข้อสอบมีทั้งหมด 3 ส่วน 2+2+6 ข้อใหญ่ ในกระดาษคำถาม 12 หน้า รวมทั้งแผ่นนี้ คะแนนรวมทั้งหมด 60 คะแนน (เก็บ 40%)
2. ให้เขียนตอบในกระดาษคำถามนี้เท่านั้น (ตอบเป็นภาษาไทยได้) และให้เขียนชื่อ เลขประจำตัว และเลขที่ใน CR58 ในทุกหน้า
3. **ไม่อนุญาตให้นำตำราและเอกสารใดๆ เข้าในห้องสอบ**
4. **ไม่อนุญาตให้ใช้เครื่องคำนวณ**
5. ห้ามการหยิบยืมสิ่งใด ๆ ทั้งสิ้น จากผู้สอบอื่น ๆ เว้นแต่เจ้าหน้าที่ควบคุมการสอบจะหยิบยืมให้
6. ห้ามนำส่วนใดส่วนหนึ่งของข้อสอบออกจากห้องสอบ
7. ผู้เข้าสอบสามารถออกจากห้องสอบได้ หลังจากผ่านการสอบไปแล้ว 45 นาที
8. เมื่อหมดเวลาสอบ ผู้เข้าสอบต้องหยุดการเขียนใดๆ ทั้งสิ้น
9. **นิสิตกระทำผิดเกี่ยวกับการสอบ ตามข้อบังคับจุฬาลงกรณ์มหาวิทยาลัย มีโทษ คือ พ้นสภาพการเป็นนิสิต หรือ ได้รับสัญลักษณ์ F ในรายวิชาที่กระทำผิด และอาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้**

**ห้ามนิสิตพกโทรศัพท์และอุปกรณ์สื่อสารไว้กับตัวระหว่างสอบ หากตรวจพบจะถือว่านิสิตกระทำผิดเกี่ยวกับการสอบ อาจต้องพ้นสภาพการเป็นนิสิต หรือ ให้ได้รับ F และอาจพิจารณาให้ถอนรายวิชาอื่นทั้งหมดที่ลงทะเบียนไว้ในภาคการศึกษานี้**

**\* ร่วมรณรงค์การกระทำผิด หรือทุจริตการสอบเป็นศูนย์ที่คณะวิศวกรรมศาสตร์ \***

**ข้าพเจ้ายอมรับในข้อกำหนดที่กล่าวมานี้ ข้าพเจ้าเป็นผู้ทำข้อสอบนี้ด้วยตนเองโดยมิได้รับการช่วยเหลือ หรือให้ความช่วยเหลือ ในการทำข้อสอบนี้**

ลงชื่อนิสิต..............................................................

วันที่........................................................................

**Part 1 Alternative Programming Model - Prolog (20 marks)**

**(If you think there is a problem with any question, please specify and explain your answer to the best of your understanding.)**

1. (12.5 marks) A vending machine sells drinks. Currently there are 2 drinks offered by the machine, Pepsi (17 Baht) and drinking water (7 Baht). **There may be other drinks in the future**. The machine accepts 10,5, and 1 Baht coins **ONLY**.

The machine works as follows:

* The machine shows which drinks are still available. (This should be shown regularly in your prolog program).
* User can insert a coin, only 1 coin at a time.
  + after each coin is inserted, the machine shows which drink(s) are available to buy, depending on the total amount the user inserted into it (the available drink must not be the one that runs out).
  + The machine can run out of each drink.
* User can also select a drink (assume user only select a named drink that the machine sells).
  + If the drink is not available, the machine warns the user, and does not do anything else.
  + Otherwise, if the money is not enough for the drink, the machine warns the user, and does not do anything else.
  + If the money is enough for the drink, and the drink is available, the machine gives the drink and changes (if there is any). Assume that the machine always has enough change. If the machine runs out of drink after this purchase, the system must show the user.

Your task is to implement a prolog system that simulates this vending machine.

The following predicates are available to call:

* isNumber(X). This predicate is true when X is a number.
* change(X,P). This predicate handles all the work for giving change, including printing all necessary messages. X is the current user’s money in the machine. P is the price of a correctly selected drink.

1. (3 marks) Design how you would store information (e.g. amount of money inserted, available drinks, etc.) using prolog.

ใช้ money(M) ในการเก็บค่าเงินที่หยอดเข้ามาแล้ว

เก็บข้อมูลเครื่องดื่มด้วย drink(Name, Cost, Amount)

Name : ชื่อของเครื่องดื่ม

Cost : ค่าใช้จ่าย

Amount : จำนวนในตู้

1. (9.5 marks) Use all your knowledge in prolog to implement predicate “operate”, which runs the vending machine. Write at the back of this paper if you need more space. “operate” must run indefinitely. Point out and implement any missing cases not mentioned in the machine description.

money(0).

addMoney(X) :- isNumber(X),

money(M),

M2 is M + X,

retract(money(M)),

assertz(money(M2)), ! .

available(X) :- drink(X,\_,Y), Y > 0.

check(X,C) :- drink(X,N,Y), Y > 0, C >= N.

show\_a() :- available(X), write(X), nl, false.

show\_available() :- show\_a(); true.

show\_c(C) :- check(X,C) ,write(X), nl, false.

show\_check(C) :- show\_c(C); true.

do\_o(X) :- X = i,

money(M),

M2 is M + 1,

retract(money(M)),

assertz(money(M2)),

show\_check(M),

write(‘insertion compleate’),

do\_o(X) :- \+ available(X), write(‘warning : drink not available.’), nl, !.

do\_o(X) :- Money(M), \+ check(X, M), write(‘warning : has not enough money.’), nl, !.

do\_o(X) :- drink(X, C, A),

money(M),

A2 is A – 1,

addMoney(-C),

assertz(drink(X, C, A2)),

retract(drink(X, C, A2)),

change(M, C).

retract(money(M)),

money(0), nl, !.

operate().

operate() :- show\_available(),

write(‘write i for coin insertion or name of drink for buying’),

input(X),

do\_o(X),

operate().

1. (7.5 marks) Write the following predicate for a list in prolog.
2. (3 marks) reverse(List,List2): true when List2 has all data from List1, but organized in a reverse order. For example, if List1 is [1,2,3,4] then List2 is [4,3,2,1].

append([], N, [N]).

append([H|T], N, Answer) :- append(T, N, A), Answer = [H|A].

reverse([],[]).

reverse([H|T],List2) :- reverse(T, A), append(A, H, List2).

1. (4.5 marks) palindrome(List): true when List is a palindrome. That is, reading data from front to back and back to front give the same result. For example, these are palindrome:

* []
* [1]
* [p,p]
* [1,2,3,2,1]
* [z,a,b,c,c,b,a,z]

do\_p([], []).

do\_p([H1|T1], [H2|T2]) :- H1 = H2, do\_p(T1,T2).

palindrome([]).

palindrome(List) :- reverse(List, List2), do\_p(List, List2).

**Part 2 Language Implementation (20 marks)**

**(If you think there is a problem with any question, please specify and explain your answer to the best of your understanding.)**

1. (10 marks) Examples of sentences in a language are shown below. Write a grammar for this language.

f()

f(a)

f(a+b)

f(a)+f(b)

f(a)+a+f(b)+...

1. (10 marks) Given the grammar below, compute its first set, follow set and draw the parsing table. The language is a simplified version of my Retro Basic.

Here is an example of the program in this language.

10 A = 1

20 PRINT A

30 GOTO 10

40 B = A + 1

pgm := line pgm | EOF

line := line\_num stmt

stmt := asgmnt | print | goto

asgmnt := id = exp

exp := term + term | term - term | term

term := id | const

print := PRINT id

goto := GOTO line\_num

id is {A..Z}

const is {1..100}

line\_num is {1..1000}

**Part 3 Programming Language Concepts (20 marks)**

**(If you think there is a problem with any question, please specify and explain your answer to the best of your understanding.)**

1. (6.5 marks) Jabba, a Java-like language, allows nested method definition. A Jabba code is given below:

public class JabbaTheHutt{

public static void main(String[] args){

int x =0;

int y =2;

public void method01(int x){

public void method2(int y){

x = x+y;

method3(x);

}

public void method3(int z){

z = z+x;

System.out.println(x + “,” +y +”,”+ z); //line1

}

z = x+y;

y = y-x;

x+=y;

method2(x);

}

method1(x+1);

System.out.println(x + “,” +y); //line2

}

}

1. (0.5 mark) Does this code need any fixing to make it compile? Please identify the fixing.
2. (3 marks) If this code uses static scope, what will be printed at line1 and line 2?
3. (3 marks) If this code uses dynamic scope, what will be printed at line1 and line 2?
4. (4 marks) A code is given below (assume that varable race is type int):

**switch** (race) {

**case** 1:

name = "Legolas ";

**break**;

**case** 2:

name = "Gimli ";

**break**;

**case** 3,5:

name = "Aragon ";

**break**;

**case** 4:

name = "Lurtz ";

**break**;

**default**:

System.***out***.println("Choice not in range, please re-run the program.");

**return**;

}

Fill in its corresponding assembly code below:

goto L6

L1:

L2:

L3:

L4:

L5: print”…”

exit

T: &L1

L6: r1 = race

if goto

if goto

r1 = r1-1

r1 = T[r1]

goto \*r1

L7:

1. (3 marks) Widening conversion of reference types in Java is the conversion of a subclass to its superclass and is carried out implicitly. On the other hand, narrowing (downcasting) is the conversion of a superclass to its subclass. It requires explicit casting and may result in runtime exception (ClassCastException).

Given the Java code below, choose the term(s) which describe what happen to each statement. The terms are 1) Widening 2) Narrowing

3) Compile OK 4) Compile error

5) Run OK 6) Runtime error.

(Put only the number 1-6 that represents the term you choose in the space provided. For example, if you choose Widening and Compile OK and run OK, put 1,3,5 in the space.)

class Animal {..}

class Cat extends Animal {..}

class Dog extends Animal {..}

…

Animal s1, s2;

Dog s3;

Cat s4;

Animal s5;

Cat s6, s7;

s1 = new Cat(); ………………………………………………………………

s2 = new Animal(); ………………………………………………………………

s4 = new Cat();

s3 = s2; ………………………………………………………………

s4 = (Cat)s2; ………………………………………………………………

s6 = s1; ………………………………………………………………

s5 = new Cat();

s7 = (Cat)s5; ………………………………………………………………

1. (2 marks) Determine what the following code (C++) prints:

class Thing { …};

…

void f(Thing\* x, Thing\* y) {

x = y;

}

…

Thing \*a = new Thing();

Thing \*b = new Thing();

f(a, b);

if (a == b) {

cout << "A" << endl;

} else {

cout << "B" << endl;

}

1. (2 marks) Determine what the following code prints:

class Thing { …};

…

void f(Thing\* &x, Thing\* &y) {

x = y;

}

…

Thing \*a = new Thing();

Thing \*b = new Thing();

f(a, b);

if (a == b) {

cout << "A" << endl;

} else {

cout << "B" << endl;

}

1. (2.5 marks) Given code in Java language (the language uses dynamic method binding). Indicate on each provided line what is being printed. If a line cannot be printed, indicate the reason (then treat the error as non-existence).

class Robot {

public void operate(){ System.out.println(“OP”); }

}

class SuperRobot extends Robot{

public void operate() { System.out.println(“OP02”); }

public void finishingMove(){ System.out.println(“YAA” };

}

class Test{

public static void main(…){

Robot x,y;

SuperRobot s;

x = new Robot();

y = new Robot();

s = new SuperRobot();

Robot r1 = s;

r1.finishingMove(); ……………………………………………………………….…………………………..

r1.operate(); ……………………………………………………………………………………………

Robot r2 = (Robot)s;

r2.operate(); ……………………………………………………………………………………………

y=x;

SuperRobot r3 = (SuperRobot) y;

r3.operate(); ………………………………………………………………………………………

s = (SuperRobot)r1;

s.operate(); ……………………………………………………………………………………….