**TCA SYSTEM DEVELOPMENT**

**Section A**

1. The ‘+’ operator can used to add two variables .eg. int x + in y.

b. The ‘+’ operator can used to concatenate two strings .eg. “hello” + “world”.

1. 0 <= n && n <= 9
2. \_\_\_
3. Syntax Error is a mistake occurred in a program due to the way you wrote the code like if the open braces of an if statement doesn’t have a close brace while a bug is problem detected while testing or developing the program.
4. A class is a template where the object will be created. It’s like a blueprint for declaring objects while an object is a member of a class which has a behaviour.
5. The coordinate system in Java’s coordinate systems differs from the common Cartesian coordinate systems in that the orientation of the y axis is upside down.  Locations near the bottom of the screen have larger y coordinates than positions near the top of the screen.
6. The stepwise refinement is the process of breaking down a Java program problem into a series of steps like putting them into functions.
7. Named constants can be use in a way that if the programmer needs to change the value throughout the program code, they only need to make one change. This can help make a program easier to maintain.
8. A predicate method are functions that returns TRUE or FALSE. They are used to check if what if put in meets our condition
9. A call is used to ask for a value from the function or a different function. An argument is a value put in a function when the function has been called. A return is used in a method to send back the value to what called it.

**Section B**

1.

import acm.program.\*;

import java.util.ArrayList;

public class LargeNum extends ConsoleProgram {

public void run() {

println("This Program find the largest integer in a list");

int NUM;

Integer MAXNUM = 0;

ArrayList<Integer> numberList = new ArrayList<>();

do {

NUM = readInt("Enter value: ");

numberList.add(NUM);

} while (NUM > 0);

for(int i = 0; i < numberList.size(); i++) {

if(numberList.get(i) > MAXNUM) {

MAXNUM = numberList.get(i);

}

}

print("Large is: " + MAXNUM);

}

}

A screenshot of a computer

Description automatically generated with medium confidence

2.

import acm.program.ConsoleProgram;

public class MATHS extends ConsoleProgram {

public void run() {

int n = 0;

n = readInt();

int count = 0;

while(n !=1) {

if (n % 2 !=0) {

println(n+"IS ODD, MAKE 3n+1:"+(3 \* n + 1));

n=(3 \* n) + 1;

} else if(n % 2 == 0) {

println(n + "IS ODD, HALF OF IT: "+(n/2));

n =n/2;

}

}

println(count +" HAS REACHED 1");

}

}

Text

Description automatically generated

**Section C**

**1.**

package moreGraphicsExamples;

import acm.graphics.\*;

import java.awt.Color;

import acm.program.\*;

public class circle extends GraphicsProgram {

private static final int WINDOW\_WIDTH = 1000;

private static final int WINDOW\_HEIGHT = 800;

public void run() {

int r1 = 500;

int r2 = 210;

setSize(WINDOW\_WIDTH, WINDOW\_HEIGHT);

GRect rect1 = new GRect(WINDOW\_WIDTH/2 - r1/2, WINDOW\_HEIGHT/2 - r1/2, r1, r2);

rect1.setColor(Color.red );

rect1.setFilled(true);

GOval oval1 = new GOval((WINDOW\_WIDTH/2 - r1/2),(WINDOW\_HEIGHT/2 - r1/2),r1,r2);

oval1.setColor(Color.green );

oval1.setFilled(true);

add(rect1);

add(oval1);

A screenshot of a computer

Description automatically generated

2.

package moreGraphicsExamples;

import java.awt.Color;

import acm.graphics.\*;

import acm.program.\*;

public class House extends GraphicsProgram {

private static final int WINDOW\_WIDTH = 1500;

private static final int WINDOW\_HEIGHT = 1000;

private static final int HOUSE\_WIDTH = 750;

private static final int HOUSE\_HEIGHT = 400;

private static final int DOOR\_HANDLE\_RADIUS = 10;

private static final int HOUSE\_W\_WIDTH = 100;

private static final int HOUSE\_W\_HEIGHT = 100;

// private static final int ROOF\_WIDTH = 60;

// private static final int ROOF\_HEIGHT = 20;

public void run() {

setSize(WINDOW\_WIDTH, WINDOW\_HEIGHT);

addHouse(getWidth() / 2, getHeight() / 2);

}

private void addHouse(double cx, double cy) {

add\_houseBody(cx, cy);

add\_doorHandle(cx, cy + HOUSE\_HEIGHT / 4);

add\_window(cx + HOUSE\_HEIGHT / 4, cy - HOUSE\_HEIGHT / 4);

add\_window2(cx - HOUSE\_HEIGHT, cy - HOUSE\_HEIGHT /4);

add\_door(cx, cy + HOUSE\_W\_WIDTH / 4);

}

private void add\_houseBody(double cx, double cy) {

double x = cx - HOUSE\_WIDTH / 2;

double y = cy - HOUSE\_HEIGHT / 2;

GRect houseBody = new GRect(x, y, HOUSE\_WIDTH, HOUSE\_HEIGHT);

add(houseBody);

}

private void add\_window(double cx, double cy) {

double x = cx + HOUSE\_W\_HEIGHT /4;

double y = cy - HOUSE\_W\_HEIGHT /4;

GRect houseWindow = new GRect(x , y, HOUSE\_W\_WIDTH, HOUSE\_W\_HEIGHT);

add(houseWindow);

}

private void add\_window2(double cx, double cy) {

double x = cx + HOUSE\_W\_HEIGHT;

double y = cy - HOUSE\_W\_HEIGHT /4;

GRect houseWindow = new GRect(x, y, HOUSE\_W\_WIDTH, HOUSE\_W\_HEIGHT);

add(houseWindow);

}

private void add\_door(double cx, double cy) {

double x = cx - HOUSE\_W\_HEIGHT;

double y = cy - HOUSE\_W\_HEIGHT /4;

GRect door = new GRect(x, y, HOUSE\_W\_HEIGHT, HOUSE\_W\_HEIGHT \*2);

add(door);

}

private void add\_doorHandle(double cx, double cy) {

double x = cx - DOOR\_HANDLE\_RADIUS;

double y = cy - DOOR\_HANDLE\_RADIUS /4;

GOval doorHandle = new GOval(x, y, DOOR\_HANDLE\_RADIUS, DOOR\_HANDLE\_RADIUS);

add(doorHandle);

}

}

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated