**Java Programming**

Come up with a few questions

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Section 1) Programming Languages**

1. What is programming? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. What type of programming is java?
   1. Muti-Paradigm
   2. Structure Oriented
   3. Object Oriented
   4. Interpreted Object Oriented
3. What are levels? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Section 2) Java Basics**

1. Give a couple examples of data types. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ , \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. What is the class that calls a math function? \_\_\_\_\_\_\_\_\_\_\_\_\_\_.LOG();
3. What data type stores logical data? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. There is only one way to program something. True False
5. How do you stop a loop?
   1. Staring at the monitor
   2. By a class interrupt
   3. Tell Condition is met
   4. Boolean Expression
6. Can you make a loop run forever? Yes No
7. What type of loop is good for arrays? While For Do,While
8. Why do you use If, Else Statements? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Section 3) Java Programming**

Reference:

**Loops**

while (*condition*) {

*// code block to be executed*

}

Example:

int i = 0;

while (i < 5) {

*// code block to be executed*

i++;

}

do {

*// code block to be executed*

}

while (condition);

Example:

int i = 0;  
do {

*// code block to be executed*

i++;

}

while (i < 5);

for (initialization; condition; increment(++)/decrement(--)) {

*// code block to be executed*

}

Example:

for (int i = 0; i < 5; i++) {

*// code block to be executed*

}

**If Else**

if (condition1) {

// block of code to be executed if condition1 is true

} else if (condition2) {

// block of code to be executed if the condition1 is false and condition2 is true

} else {

// block of code to be executed if the condition1 is false and condition2 is false

}

Example:

int time = 20;

if (time < 18) {

System.out.println("Good day.");

} else {

System.out.println("Good evening.");

}

variable = (condition) ? expressionTrue : expressionFalse;

Example:

int time = 20;

String result = (time < 18) ? "Good day." : "Good evening.";

System.out.println(result);

**Switch/Case Structure**

switch(expression) {

case x:

// code block

break;

case y:

// code block

break;

default:

// code block

}

Example:

int day = 4;

switch(day) {

case 6:

System.out.println("Today is Saturday");

break;

case 7:

System.out.println("Today is Sunday");

break;

default:

System.out.println("Looking forward to the Weekend");

}

// Outputs "Looking forward to the Weekend"

**Setting Constants**

int myNum = 5; // Integer (whole number)

double myDouble = 5.99f; // Double (decimal number)

float myFloatNum = 5.99f; // Floating point number

char myLetter = 'D'; // Character

boolean myBool = true; // Boolean

String myText = "Hello"; // String

String[] cars = {"Volvo", "BMW", "Ford", "Mazda"}; //String Array

int[] myNum = {10, 20, 30, 40}; //Integer Array

**String Functions**

string.method(Settings1, Settings3, ect.);

substring(int beginIndex, int endIndex) - Extracts the characters from a string, beginning at a specified start position, and through the specified number of character.

concat(String str) - Appends a string to the end of another string.

indexOf(int ch) - Returns the position of the first found occurrence of specified characters in a string.

length() - Returns the length of a specified string as an integer.

Go to: <https://www.w3schools.com/java/java_ref_string.asp> to find out more functions

**Number Functions**

Math.method(Settings1, Settings3, ect.);

round(x) - Returns the value of x rounded to its nearest integer

random() - Returns a random number between 0 and 1

max(x, y) - Returns the number with the highest value

min(x, y) - Returns the number with the lowest value

abs(x) - Returns the absolute value of x

sqrt(x) - Returns the square root of x

Go to: <https://www.w3schools.com/java/java_ref_math.asp> to find out more functions

**Section 4) Robot Programming Part – 1**

1. How many communication protocols does the RoboRIO have? \_\_\_\_\_\_
2. What does PWM mean? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. What do we use to run motors? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Why do we use sensors? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. What sensors can measure distance?
   1. Hall Effect
   2. LIDAR
   3. Gyroscopes
   4. Camera
6. What type of operator control works with one x and y axis?
   1. Arcade Control
   2. Tank Control
   3. Omi-Directional Control
   4. Curvature Control
7. How many type of drive systems are there? \_\_\_\_\_\_
8. What do we use to communicate with the robot?\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Section 5) Robot Programming Part- 2**

Reference:

**robotInit() - This function is run when the robot is first started up and should be used for any initialization code.**

**robotPeriodic() - This function is called every robot packet, no matter the mode.**

**autonomousInit() – This function is called before autonomous is started.**

**autonomousPeriodic() – This function is where autonomous mode is ran.**

**teleopInit() - This function is called before teleop is started.**

**teleopPeriodic() – This function is where teleop is ran**

**testPeriodic() – This function is for code that needs to be tested.**

**First Example Review:**

package frc.robot;

import edu.wpi.first.wpilibj.Joystick;

import edu.wpi.first.wpilibj.PWMVictorSPX;

import edu.wpi.first.wpilibj.InteractiveRobot;

import edu.wpi.first.wpilibj.drive.DifferentialDrive;

import edu.wpi.first.wpilibj.Spark;

public class Robot extends InteractiveRobot {

private DifferentialDrive m\_myRobot;

private Joystick m\_leftStick;

private Joystick m\_rightStick;

private Spark m\_leftMotor = new Spark(0);

private Spark m\_rightMotor = new Spark(1);

@Override

public void robotInit() {

m\_myRobot = new DifferentialDrive(m\_leftMotor, m\_rightMotor);

m\_leftStick = new Joystick(0);

m\_rightStick = new Joystick(1);

}

@Override

public void teleopPeriodic() {

m\_myRobot.tankDrive(m\_leftStick.getY(), m\_rightStick.getY());

}

}

**Information:**

[**https://first.wpi.edu/FRC/roborio/release/docs/java/**](https://first.wpi.edu/FRC/roborio/release/docs/java/)