

## Lab Practicum

Purpose: You will be designing, creating, and debugging program software that implements input validation, decision making, and repetition. The software program will be designed using pseudocode and flowcharts, and then implemented in the Python programming language.

### Deliverables, Requirements, and Timeline

#### 1. Choosing a Project Topic:

1. Choose a project topic. Either create your own or choose from one of the following topics:

##### 1. **Programming a Robot Battlebot**

Write a program for a Robot Battlebot that satisfies these conditions:

1. The battlebot should have the following command menu selections: Fire a Weapon, Move Forward, Move Backward, Exit
2. If the Fire Weapon command is given, the user should be shown the distance to the target and prompted for whether they want to fire or not.

If they choose to fire, the following messages (or something similar) is to be displayed:

If the opponent is within 20 feet, a message should be displayed that the opponent is destroyed; if it is within 40 feet, the message should say it is partially disabled; and if it is over 40 feet away, the message should say it is unharmed. The battlebot has enough ammunition to fire its weapon five times in total. When it is out of ammunition, it should display a message that it cannot fire anymore when the Fire Weapon command is given.

2            ~ IT104 – Intro to Programming ~  
Prof. Halton 678-687-6104 mhalton@itt-tech.edu  
              :: Lab Practicum ::

3. • If the Move Forward command is given, the user should be prompted to enter how far they want to go  
The program should randomly place an obstacle in the path. If the obstacle is closer than the desired travel distance, the user should be informed and given the opportunity to move only the available distance or not to move forward. If the battlebot is not blocked, it should display a message that it has move the entire distance and state the distance. The fuel quantity must be monitored, as the battlebot starts with only 200 ft/units of fuel.
4. • The Move Backward command should work the same way as the Move Forward command.
5. • The Battlebot should keep track of how far it travels. It has enough fuel to travel 200 feet in total. If it runs out of fuel, it should display a message that it is out of fuel when the Move Forward or Move Backward command is given.

**4. Create the Pseudocode:**

1. You should consider the following accumulator variables in designing your pseudocode:
  - a) Programming a Robot Battlebot  
ammunition = 5 (initially)  
fuel = 200 (initially)
2. Program code for the menu selections should be placed inside modules named as follows:
  - a) Programming a Robot Battlebot

:: Lab Practicum ::

```
fireWeapon( )  
moveForward( )  
moveBackward( )
```

3. All projects should implement a While loop in their code to display the command selection menu. The While loop should use a variable called choice to capture the user's menu selection. The While loop should test to see what value the variable choice has to determine which menu option was selected. Invalid selections should receive a message to input selection again. You can use the following pseudocode as an outline of the selection menu loop:

**a) Programming a Robot Battlebot**

```
//variable declarations:  
Declare Integer choice = 0  
Declare Integer ammunition = 5  
Declare Real fuel = 200.0  
  
//main selection menu  
While choice != 4  
    //display menu  
    Display "Menu Selections:"  
    Display "1 -Fire Weapon"  
    Display "2 -Move Forward"  
    Display "3 -Move Backward"  
    Display "4 -Exit"  
    Display "Enter your selection"
```

4 ~ IT104 – Intro to Programming ~  
Prof. Halton 678-687-6104 mhalton@itt-tech.edu  
:: Lab Practicum ::

```
Input choice
//check menu selection
If choice == 1 Then
    fireWeapon( )
Else If choice == 2 Then
    moveForward( )
Else If choice == 3 Then
    moveBackward( )
Else If choice == 4 Then
    Display “Goodbye!”
Else
    Display “Invalid input –please try again.”
End If
End While
```

4. Create the logic chart of the pseudocode using Raptor.  
or create a flowchart using Visio, OpenDraw or some other graphical application
5. Translate the pseudocode into Python program code.
6. Test that the program will only accept valid user input, and not merely crash.

Deliverables: You will submit a Word document containing the following:

- Cover page with Project Name, Team Members, Instructor Name, and Course Number
- Pseudocode for the program

:: Lab Practicum ::

- Flowchart for the program from Visio  
or Logic Chart from Raptor
- Python program code
- Working program demonstration