Nutan Maharashtra Vidya Prasarak Mandal's

NUTAN MAHARASHTRA INSTITUTE OF ENGINEERING AND TECHNOLOGY

Department of
Computer Engineering

LABORATORY MANUAL

SUBJECT: LABORATORY PRACTICE-I

[SUBJECT CODE: 310248]

CLASS: T.E. COMP

YEAR: 2021-2022

Vision and Mission of the Institute

1. **Vision of the Institute-**To be a notable institution for providing quality technical education, ensuring ethical, moral, and holistic development of students.
2. **Mission of the Institute**- To nurture engineering graduates with highest technical competence, professionalism and problem solving skills to serve the needs of industry and society.

Vision and Mission of the Department

Department Vision:

Imbibing Quality Technical Education and Overall Development by Endowing Students with Societal and Ethical skills in Computer Engineers

Department Mission:

1. To impart engineering knowledge and skills by adopting effective teaching learning processes.
2. To develop professional, entrepreneurial & research competencies encompassing continuous intellectual growth
3. To produce educated students to exhibit societal and ethical responsibilities in the working environment.

Program Educational Objectives (PEOs):

PEO1: To produce globally competent graduates having Excellent fundamentals, domain knowledge, updated with modern technology to provide effective solutions for computer engineering problems
PEO2: To prepare the graduates to work as a committed professional with professional ethics and values, sense of responsibilities, understanding of legal, safety, health, societal, cultural and environmental issues.
PEO3: To prepare committed and motivated graduates with research attitude, lifelong learning, investigative approach, and multidisciplinary thinking
PEO4: To produce the graduates with well-built managerial and communication skills to work effectively as individual as well as in teams

Program Outcomes

1. **Engineering knowledge**:

Graduates can apply the knowledge of mathematics, science, engineering fundamentals and an engineering specialization to Civil Engineering related problems.

2. **Problem analysis**:

An ability to identify, formulate, review research literature, and analyse Civil engineering problems reaching substantiated conclusions using principles of mathematics and engineering sciences.

3. **Design/development of solutions**:

An ability to plan, analyse, design, and implement engineering problems and design system components or processes to meet the specified needs.

4. **Conduct investigations of complex problems**:

An ability to use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**:

An ability to apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. **The engineer and society**:

An ability to apply contextual knowledge to assess societal, legal issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**:

An ability to understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**:

An ability to apply ethical principles and commit to professional ethics and responsibilities and norms ofthe engineering practice.

9. **Individual and teamwork**:

An ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings to accomplish a common goal.

10. **Communication**:

An ability to communicate effectively on engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation and make effective presentations.

11. **Project management and finance**:

Ability to demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**:

An ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO 1 - Employ knowledge to write programs and design algorithms to integrate them with the hardware/software products in the domains of embedded systems, data Science, networking and web technology.

PSO 2 - Apply standard practices and strategies in project development using open-ended programming environments to create innovative career paths to be an entrepreneur, and a zest for higher studies.

Course Outcomes (CO)

Systems Programming and Operating System	
CO 1	Implement language translators
CO 2	Use tools like LEX and YACC
CO 3	Implement internals and functionalities of Operating System
Internet of Things and Embedded Systems	
CO 4	Design IoT and Embedded Systems based application
CO 5	Develop smart applications using IoT
CO 6	Develop IoT applications based on cloud environment
Software Project Management	
CO 4	Apply Software Project Management tools
CO 5	Implement software project planning and scheduling
CO 6	Analyse staffing in software project

List of Experiments with Mapping

Sr. No.	Name of the Experiment	CO mapping	Level of CO mapping
Part I: Systems Programming and Operating System			
1.	Group A I: Systems Programming and Operating Systems, Design suitable data structures and implement pass1 and pass2 of a two pass assembler for pseudo-machine. Implementation should consist of a few instructions from each category and few assembler directives. The output of pass1 (intermediate code file and symbol table) should be input for pass2	CO1, CO3	2
2.	Group A Design suitable data structures and implement Pass-I of a two pass macro processor using OOP features in Java/C++. The output of Pass-I (MNT, MDT, ALA & Intermediate code file without any macro definitions) should be input for Pass-II.	CO2, CO3	3
3.	**Group B** Write a program to simulate CPU Scheduling Algorithms: FCFS, SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive).	CO3	3
4 .	**Group B** Write a program to simulate Memory placement strategies – best fit, first fit, next fit and worst fit.	CO3	3
5.	**Group B** Write a program to simulate Page replacement algorithm.	CO3	3
Part II : Elective I			
Internet of Things and Embedded Systems			
6	Understanding the connectivity of Raspberry-Pi / Adriano with IR sensor. Write an application to detect obstacle and notify user using LEDs.	CO4, CO5	3
7	Understanding the connectivity of Raspberry-Pi /Beagle/ Arduino UNO circuit board circuit with temperature sensor. Write an application to read the environment temperature. If temperature crosses a threshold value, generate alerts using LEDs.	CO4	2

Software Project Management			
8	Create Project Plan ▪ Specify project name and start (or finish) date. ▪ Identify and define project tasks. ▪ Define duration for each project task. ▪ Define milestones in the plan ▪ Define dependency between tasks ▪ Define project calendar. ▪ Define project resources and specify resource type ▪ Assign resources against each task and baseline the project plan	CO4,CO5	3
9	Execute and Monitor Project Plan ▪ Update % Complete with current task status. ▪ Review the status of each task. ▪ Compare Planned vs Actual Status ▪ Review the status of Critical Path ▪ Review resources assignation status	CO5, CO6	2

Rubrics for Evaluation

Sr. No	Evaluation Criteria	Marks foreach Criteria	Rubrics
1	Timely submission	5 or 10	Punctuality reflects the work ethics. Students should reflectthat work ethics by completing the lab assignments and reports in a timely manner .
2	Journal Presentation	5 or 10	Students are expected to prepare the journal. The journal presentation of the course should be complete, clear, and understandable.
3	Performance	5 or 10	After performance, the students should have good knowledge of the experiment.
4	Understanding	5 or 10	The student should be able to explain methodology used fordesigning and developing the program/solution. Student should clearly understand the purpose of the assignment and its outcome.
5	Oral	5 or 10	The student should be able to answer the questions related to the lab assignments.

Part I: Systems Programming and Operating System
Assignment No. 01

Problem Statement: Design suitable data structures and implement pass1 and pass2 of a two pass assembler for pseudo-machine. Implementation should consist of a few instructions from each category and few assembler directives. The output of pass1 (intermediate code file and symbol table) should be input for pass2

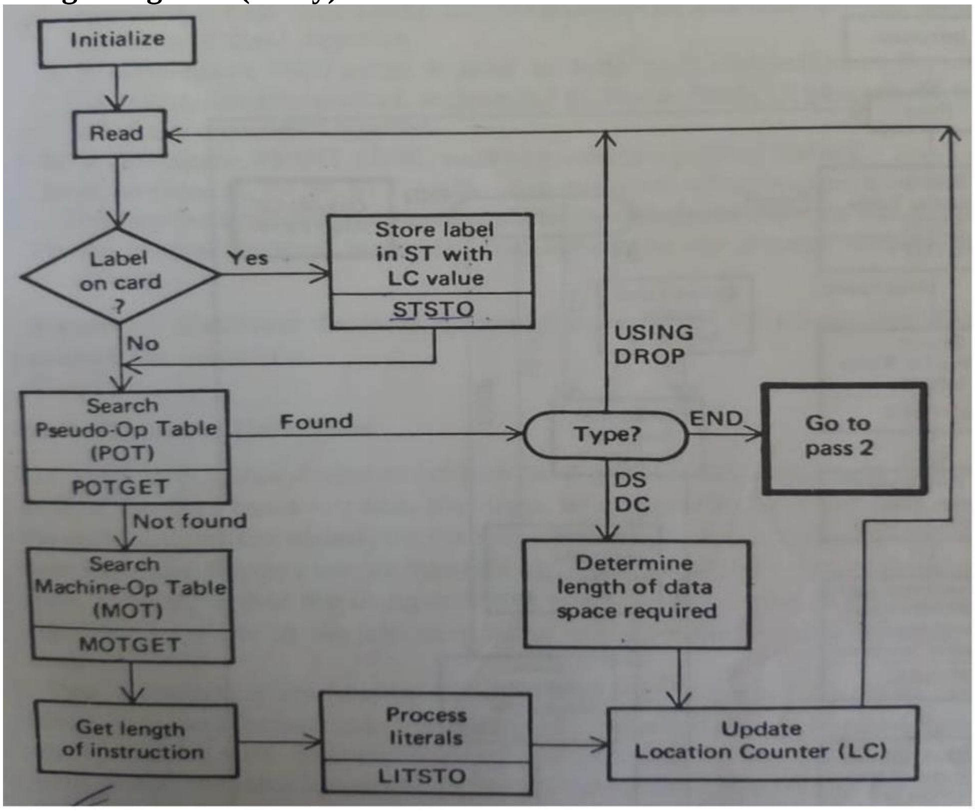
Objectives:

1. To learn systems programming tools
2. Implement language translator

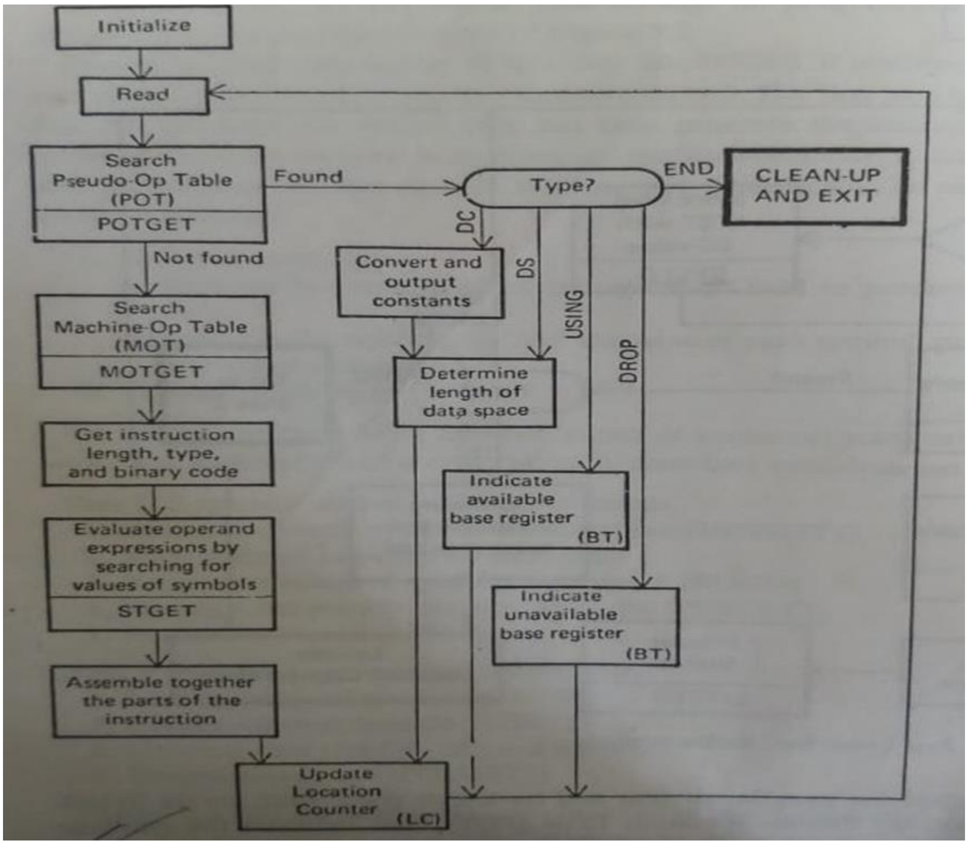
Theory:

1. Assembler is a low level translator which translates source code to machine code
2. It works in two phases : analysis phase and synthesis phase
3. In analysis phase, source assembly code is analysed to generate some intermediate data structures
4. In synthesis phase, machine code is generated
5. There are well defined system level standard algorithms to design the assembler as a two pass assembly, namely PassI and PassII algorithms of assembler
6. PassI takes the source code in assembly language as input and generates intermediate data structures.
7. PassII takes the intermediate data structures generated by PassI as input and generates machine code.

Design diagrams (if any): Flow chart for Pass-1


Initialize
Read
Label on card ?
Yes
Store label in ST with LC value
STSTO
No
USING
DROP
Search Pseudo-Op Table (POT)
POTGET
Found
Type?
END
Go to pass 2
DS
DC
Not found
Search Machine-Op Table (MOT)
MOTGET
Determine length of data space required
Get length of instruction
Process literals
LITSTO
Update Location Counter (LC)

Flow chart for Pass-2


Initialize
Read
Search Pseudo-Op Table (POT)
POTGET
Found
Type?
END
CLEAN-UP AND EXIT
DC
DS
USING
DROP
Not found
Convert and output constants
Search Machine-Op Table (MOT)
MOTGET
Determine length of data space
Get instruction length, type, and binary code
Indicate available base register (BT)
Evaluate operand expressions by searching for values of symbols
STGET
Indicate unavailable base register (BT)
Assemble together the parts of the instruction
Update Location Counter (LC)

Input for PassI of assembler:
1. Assembly language program in hypothetical language as per the Author Dhamdhere
2. OPTB
3. Condition code table
4. Register table

Input for PassII of assembler:
1. IC
2. SYMTAB
3. LITTAB
4. POOLTAB
5. OPTAB

Output of PassI of assembler:
1. IC
2. SYMTAB
3. LITTAB
4. POOLTAB

Output of PassII of assembler:
Machine code

Software Requirement:
Operating System recommended: - 64-bit Open source Linux or its derivative
Programming tools recommended: - Eclipse IDE

Hardware Requirement: I3 and I5 machines

Frequently Asked Questions:

1. What is two pass assembler?
2. What is the significance of symbol table?
3. Explain the assembler directives EQU, ORIGIN.
4. Explain the assembler directives START, END, LTORG.
5. What is the use of POOLTAB and LITTAB?
6. How literals are handled in pass I?
7. What are the tasks done in Pass I?
8. How error handling is done in pass I?
9. Which variant is used in implementation? Why?
10. Which intermediate data structures are designed and implemented in PassI?
11. What is the format of a machine code generated in PassII?
12. What is forward reference? How it is resolved by assembler?
13. How error handling is done in pass II?
14. What is the difference between IS, DL and AD?
15. What are the tasks done in Pass II?

Conclusion:

1. Input assembly language program is processed by applying PassI algorithm of assembler and intermediate data structures, SYMTAB, LITTAB, POOLTAB, IC, are generated.
2. The intermediate data structures generated in PassI of assembler are given as input to the PassII of assembler, processed by applying PassII algorithm of assembler and machine code is generated
3. Successfully implemented pass1 and pass2 of a two pass assembler for pseudo-machine

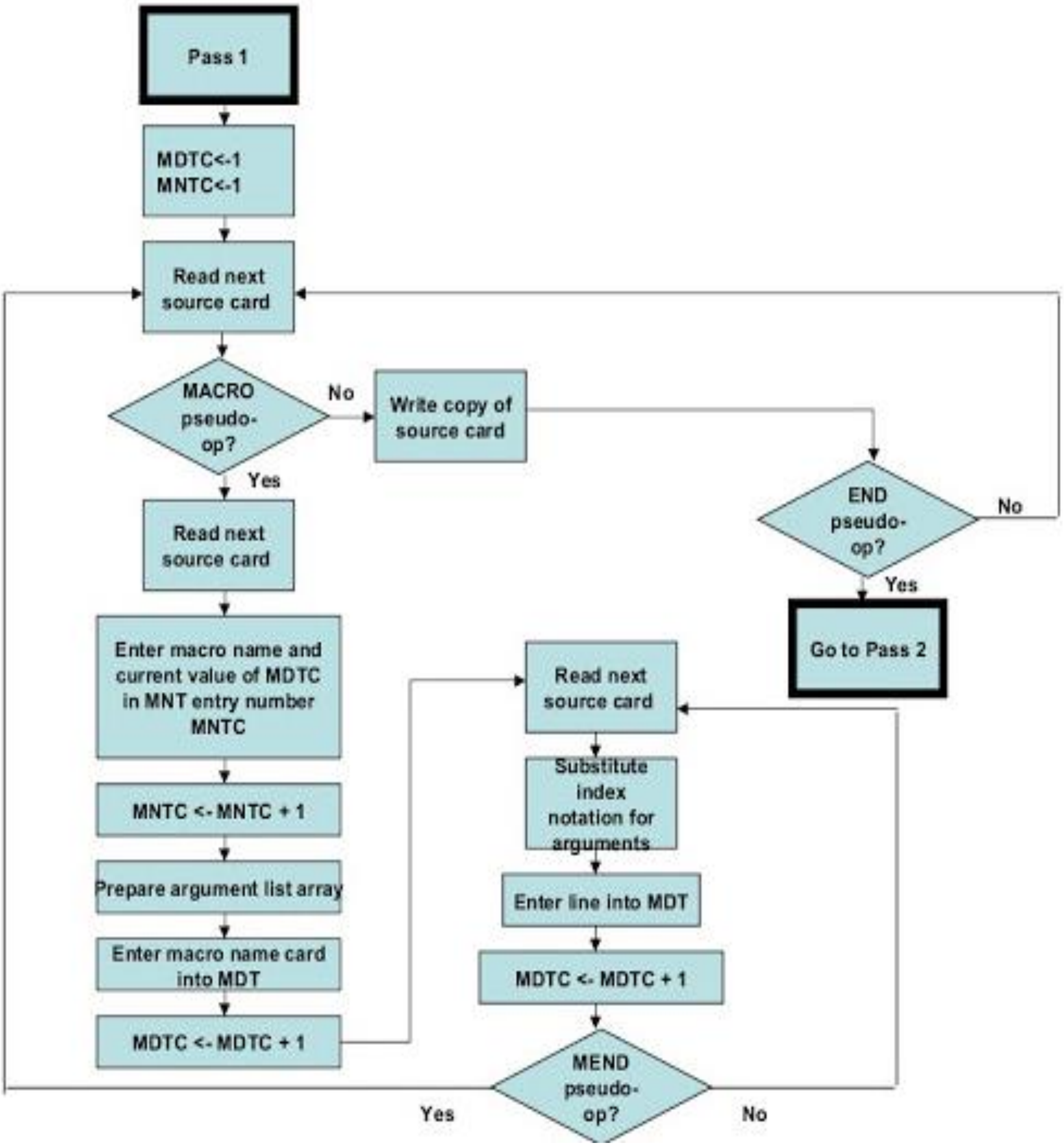
Assignment No. 02

Problem Statement: Design suitable data structures and implement Pass-I of a two pass macro processor using OOP features in Java/C++. The output of Pass-I (MNT, MDT, ALA & Intermediate code file without any macro definitions) should be input for Pass-II.

Objectives:

1. To identify and design different data structure used in macro-processor implementation
2. To apply knowledge in implementation of two pass microprocessor.

Algorithm/Flowchart:



Design diagrams (if any):

1. Flow Chart for Pass1 and Pass2
2.

Input:

Small assembly language program with macros written in file input.asm.

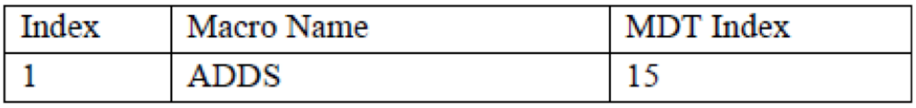
```
            MACRO
&lab    ADDS &arg1,&arg2
&lab    L 1, &arg1
            A 1, &arg2
            MEND
PROG START 0
           BALR 15,0
           USING *,15
LAB ADDS DATA1, DATA2
           ST 4,1
DATA1 DC F’3’
DATA2 DC F’4’
           END
```

Output:

Assembly language program without macro definition but with macro call.

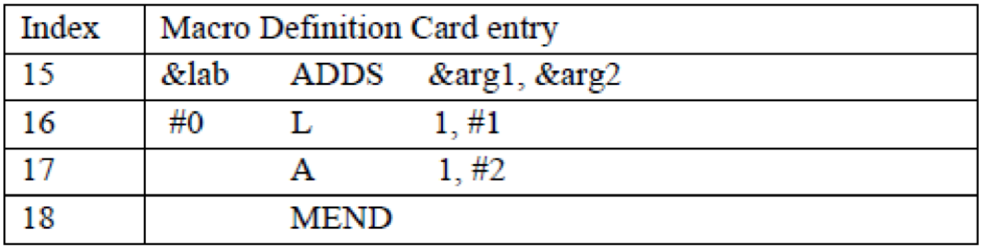
Note: Follow the following templates during implementation

Macro Name Table (MNT) :



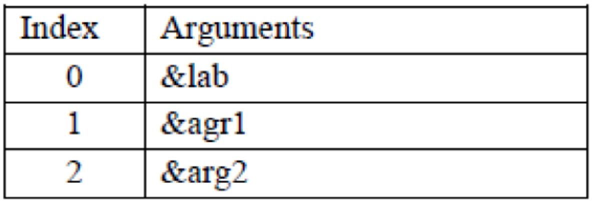
Index	Macro Name	MDT Index
1	ADDS	15

Macro Definition Table (MDT) :



Index	Macro Definition Card entry
15	&lab ADDS &arg1, &arg2
16	#0 L 1, #1
17	A 1, #2
18	MEND

Argument List Array (ALA) :



Index	Arguments
0	&lab
1	&agr1
2	&arg2

Test Cases:

1. Check macro end not found.
2. Duplicate macro name found.
3. Check program output by changing macro name and parameter list.
4. Handle label in macro definition.
5. Handle multiple macro definitions and calls

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement: N/A

Frequently Asked Questions:

1. Define macro?
2. Define purpose of pass-1 of two pass macro processor
3. List out types of macro arguments
4. What is the use of MDT-index field in MNT?
5. What we store in ALA?

Conclusion: We have successfully completed implementation of Pass-I of macro processor.

Assignment No. 03

Problem Statement: Write a program to simulate CPU Scheduling Algorithms: FCFS, SJF (Preemptive), Priority (Non-Preemptive) and Round Robin (Preemptive).

Objectives:

1. To study the process management and various scheduling policies viz.Preemptive and Non preemptive.
2. To study and analyze different scheduling algorithms.

Input:

1. Enter the number of processes:
2. Enter burst time and arrival time of each process

Theory:

FCFS: Given n processes with their burst times, the task is to find average waiting time and average turn around time using FCFS scheduling algorithm.
First in, first out (FIFO), also known as first come, first served (FCFS), is the simplest scheduling algorithm. FIFO simply queues processes in the order that they arrive in the ready queue.
In this, the process that comes first will be executed first and next process starts only after the previous gets fully executed.
Here we are considering that arrival time for all processes is 0.

How to compute below times in Round Robin using a program?

Completion Time: Time at which process completes its execution.
Turn Around Time: Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time
Waiting Time(W.T): Time Difference between turn around time and burst time.
Waiting Time = Turn Around Time – Burst Time

SJF: **Shortest Job First (SJF) Scheduling**
Till now, we were scheduling the processes according to their arrival time (in FCFS scheduling). However, SJF scheduling algorithm, schedules the processes according to their burst time.

In SJF scheduling, the process with the lowest burst time, among the list of available processes in the ready queue, is going to be scheduled next.

However, it is very difficult to predict the burst time needed for a process hence this algorithm is very difficult to implement in the system.

Advantages of SJF

1. Maximum throughput
2. Minimum average waiting and turnaround time

Disadvantages of SJF

1. May suffer with the problem of starvation

2. It is not implementable because the exact Burst time for a process can't be known in advance.

Round Robin Scheduling Algorithm

Round Robin scheduling algorithm is one of the most popular scheduling algorithm which can actually be implemented in most of the operating systems. This is the **preemptive version** of first come first serve scheduling. The Algorithm focuses on Time Sharing. In this algorithm, every process gets executed in a **cyclic way**. A certain time slice is defined in the system which is called time **quantum**. Each process present in the ready queue is assigned the CPU for that time quantum, if the execution of the process is completed during that time then the process will **terminate** else the process will go back to the **ready queue** and waits for the next turn to complete the execution.

Output:

1. Compute Waiting time, turnaround time, average waiting time, average turnaround time and throughput.

For each algorithm display result as follows:

Proces s	Burst Time	Arrival Time	Waiting Time	Turnaround Time
P1				
P2				
P3				
-				

Calculate

Average waiting time=
Average turnaround time=
Throughput=

Test Cases:

1. Check arrival time of all process should not be same.

Software Requirement:

1. Fedora
2. Eclipse
3. JDK

Hardware Requirement: for simulation no dependency

Frequently Asked Questions:

i) What are the types of CPU scheduler?

ii) What is the difference between long and short term scheduling?

iii) Logic of program?

iv) What is preemptive and non-preemptive scheduling?

v) What are types of scheduling algorithms?

vi) Why Priority scheduling may cause low-priority processes to starve?

vii) What are the goals of scheduling?

viii) Define the difference between preemptive and non-preemptive scheduling.

ix) Which scheduling algorithm is best? Why?

Conclusion: Successfully implemented CPU scheduling algorithims

Assignment No. 04

Problem Statement:

Write a Java/C++ program to simulate memory placement strategies

1. First Fit
2. Best Fit
3. Worst Fit

Objectives:

1. To acquire knowledge memory placement strategies
2. To be able to implement memory placement strategies

Theory:

- **First Fit:**

  First Fit algorithm scans the linked list and whenever it finds the first big enough hole to store a process, it stops scanning and load the process into that hole. This procedure produces two partitions. Out of them, one partition will be a hole while the other partition will store the process.

  First Fit algorithm maintains the linked list according to the increasing order of starting index. This is the simplest to implement among all the algorithms and produces bigger holes as compare to the other algorithms.

- **Best Fit:**

  The Best Fit algorithm tries to find out the smallest hole possible in the list that can accommodate the size requirement of the process.

  Using Best Fit has some disadvantages.

  1. It is slower because it scans the entire list every time and tries to find out the smallest hole which can satisfy the requirement the process.

  2. Due to the fact that the difference between the whole size and the process size is very small, the holes produced will be as small as it cannot be used to load any process and therefore it remains useless. Despite of the fact that the name of the algorithm is best fit, It is not the best algorithm among all.

- **Worst Fit:**

  The worst fit algorithm scans the entire list every time and tries to find out the biggest hole in the list which can fulfill the requirement of the process.

Despite of the fact that this algorithm produces the larger holes to load the other processes, this is not the better approach due to the fact that it is slower because it searches the entire list every time again and again.

Algorithm/ Implementation:

1. First Fit Algorithm/ Implementation

```
1- Input memory blocks with size and processes with size.
2- Initialize all memory blocks as free.
3- Start by picking each process and check if it can be assigned to current block.
4- If size-of-process <= size-of-block if yes then assign and check for next process.
5- If not then keep checking the further blocks.
```

2. Best Fit Algorithm/ Implementation

```
1- Input memory blocks and processes with sizes.
2- Initialize all memory blocks as free.
3- Start by picking each process and find the minimum block size that can be assigned to
current process i.e., find min(bockSize[1], blockSize[2],.....blockSize[n]) >
processSize[current], if found then assign it to the current process.
5- If not then leave that process and keep checking the further processes.
```

3. Worst Fit Algorithm/Pseudo code

```
1- Input memory blocks and processes with sizes.
2- Initialize all memory blocks as free.
3- Start by picking each process and find the maximum block size that can be assigned to
   current process i.e., find max(bockSize[1], blockSize[2],.....blockSize[n]) >
   processSize[current], if found then assign it to the current process.
5- If not then leave that process and keep checking the further processes.
```

Design diagrams (if any):

1. Class diagram

Input:

No. of jobs (js) & No. of blocks (bs)
Job size of all jobs & Block size of all blocks
For Example:
js=4
bs=5
block[]   = {100, 500, 200, 300, 600};
jobs[] = {212, 417, 112, 426};

Output:

Sample output of Worst Fit algorithm (same way generate o/p for other algorithms)-

```
Process No.   Process Size   Block no.

     1             212           5

     2             417           2

     3             112           4

     4             426          Not Allocated
```

Instructions: not specific

Test Cases:

Software Requirement:

1. Eclipse IDE
2. Java

Hardware Requirement: Not specific

Frequently Asked Questions:

1. Which algorithm is best and why?
2. Need of allocating blocks to jobs?
3. What is the time taken by each algorithm for execution?

Conclusion: successfully implemented simulation of memory placement strategies.

Assignment No. 05

Problem Statement: Write a Java Program (using OOP features) to implement paging simulation using 1. FIFO 2. Least Recently Used (LRU) 3. Optimal algorithm

Objectives:

1. To study page replacement policies to understand memory management.
2. To understand efficient frame management using replacement policies.

Theory:

CONCEPT OF PAGE REPLACEMENT:

1. Page Fault: Absence of page when referenced in main memory during paging leads to a page fault.

2. Page Replacement: Replacement of already existing page from main memory by the required new page is called as page replacement. And the techniques used for it are called as page replacement algorithms.

NEED OF PAGE REPLACEMENT:

Page replacement is used primarily for the virtual memory management because in virtual memory paging system principal issue is replacement i.e. which page is to be removed so as to bring in the new page, thus the use of the page replacement algorithms. Demand paging is the technique used to increase system throughput. To implement demand paging page replacement is primary requirement. If a system has better page replacement technique it improves demand paging which in turn drastically yields system performance gains.

PAGE REPLACEMENT POLICIES:

1. Determine which page to be removed from main memory.

2. Find a free frame.

   1) If a frame is found use it

   2) if no free frame found, use page replacement algorithm to select a victim frame.

   3) Write the victim page to the disk.

3. Read the desired page into the new free frame, change the page and frame tables.

4. Restart the user process.

PAGE REPLACEMENT ALGORITHMS:

1. FIFO

This is the simplest page replacement algorithm. In this algorithm, the operating system keeps track of all pages in the memory in a queue, the oldest page is in the front of the queue. When a page needs to be replaced page in the front of the queue is selected for removal.

2. OPTIMAL PAGE REPLACEMENT ALGORITHM: Replace the page that will not be used for longest period of time as compared to the other pages in main memory. An optimal page replacement algorithm has lowest page fault rate of all algorithm. It is called as OPT or MIN.

ADVANTAGE:

1) This algorithm guarantees the lowest possible page-fault rate for a fixed no. of frames.

DISADVANTAGE:

1) The optimal page replacement algorithm is very difficult to implement, as it requires the knowledge of reference strings i.e. strings of memory references.

3. LEAST RECENTLY USED (LRU): LRU algorithm uses the time of the page’s last usage. It uses the recent past as an approximation of the near future, then we can replace the page that has not been used for the longest period of the time i.e. the page having larger idle time is replaced.

ADVANTAGE:

1) The LRU policy is often used for page replacement and is considered to be good.

DISADVANTAGES:

1) It is very difficult to implement.

2) Requires substantial hardware assistance.

3) The problematic determination of the order for the frames defined by the time of last usage

Algorithm:

1. FIFO :

1. Start the process

2. Read number of pages n

3. Read number of pages no

4. Read page numbers into an array a[i]

5. Initialize avail[i]=0 .to check page hit

6. Replace the page with circular queue, while re-placing check page availability in the frame Place avail[i]=1 if page is placed in the frame Count page faults

7. Print the results.

8. Stop the process.

2. LEAST RECENTLY USED

1.Start the process

2. Declare the size

3. Get the number of pages to be inserted

4. Get the value

5. Declare counter and stack

6. Select the least recently used page by counter value

7. Stack them according the selection.

8. Display the values

9. Stop the process

3. OPTIMAL ALGORTHIM:

1. Start Program

2. Read Number Of Pages And Frames

3.Read Each Page Value

4. Search For Page In The Frames

5. If Not Available Allocate Free Frame

6. If No Frames Is Free Replace The Page With The Page That Is Least Used

7.Print Page Number Of Page Faults

8.Stop process.

Input:

1. Number of frames

2. Number of pages

3. Page sequence

Output:

1. Sequence of allocation of pages in frames (for each algorithm)

2. Cache hit and cache miss ratio.

Software Requirement:

1. Eclipse
2. JDK

Hardware Requirement: for simulation no dependency

Frequently Asked Questions:

1. What is virtual memory?
2. Explain working of LRU page replacement algorithm
3. Explain working of OPTIMAL page replacement algorithm
4. Which Page replacement algorithm is best?
5. Explain what is Belody’s Anomaly?
6. Explain the scenario in which page replacement algorithm is used?
7. Explain what is page fault?
8. Explain what is paging scheme?
9. Explain what is counting based page replacement algorithms?

Conclusion: Successfully implemented all page replacement policies.

Part II: Elective I

Internet of Things and Embedded Systems

Assignment No. 6

Problem Statement: Understanding the connectivity of Raspberry-Pi / Adriano with IR sensor. Write an application to detect obstacle and notify user using LED.

Objectives:

1. To study about Raspberry-Pi kit and its component.
2. TO study about Adriano kit and its component.
3. To Learn the interfacing of IR sensor with Arduino UNO

Outcome:

After completion of this assignment students will be able to understand the connectivity, of Raspberry-Pi and Adriano UNO with IR sensor.

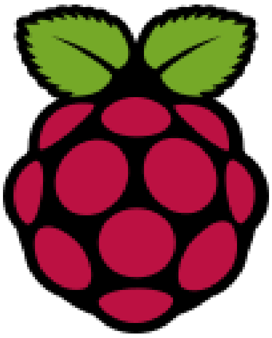
Software & Hardware Requirements:

1. Operating System: Windows (XP/Vista/7/10)
2. Software: Arduino IDE 1.8.3
3. Hardware: Raspberry -pi, Arduino UNO, IR sensor, Patch Cords, USB cable type A/B.

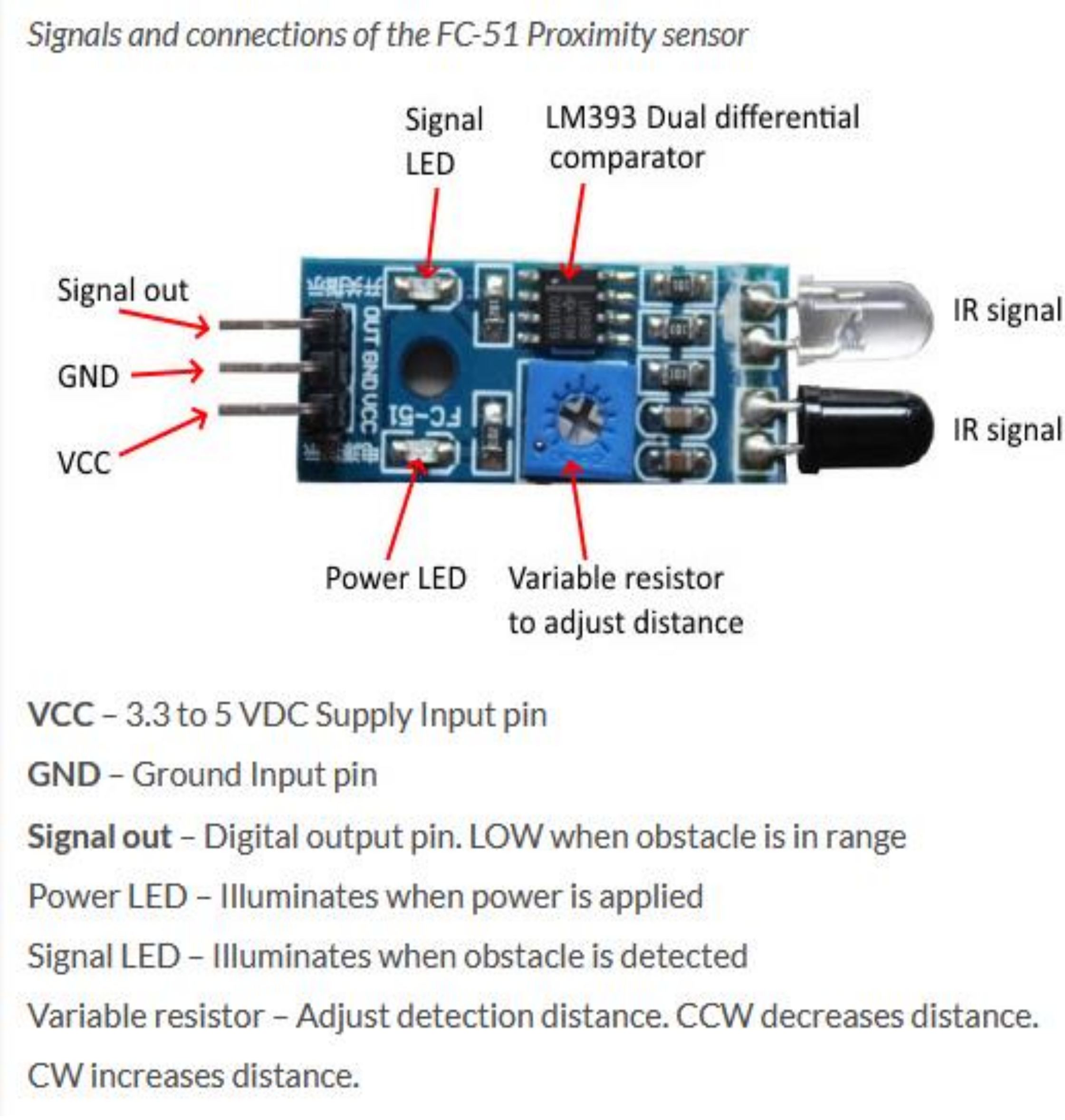
Theory: What is Arduino?

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

What is Raspberry Pi (/paɪ/) is a series of small single-board computers (SBCs) developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries. The original model became more popular than anticipated, selling outside its target market for uses such as robotics. It is widely used in many areas, such as for weather monitoring, because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of HDMI and USB devices.



IR Sensor: An infrared sensor (IR sensor) is a radiation-sensitive optoelectronic component with a spectral sensitivity in the infrared wavelength range 780 nm … 50 µm. IR sensors are now widely used in motion detectors, which are used in building services to switch on lamps or in alarm systems to detect unwelcome guests.



Signals and connections of the FC-51 Proximity sensor

VCC – 3.3 to 5 VDC Supply Input pin

GND – Ground Input pin

Signal out – Digital output pin. LOW when obstacle is in range

Power LED – Illuminates when power is applied

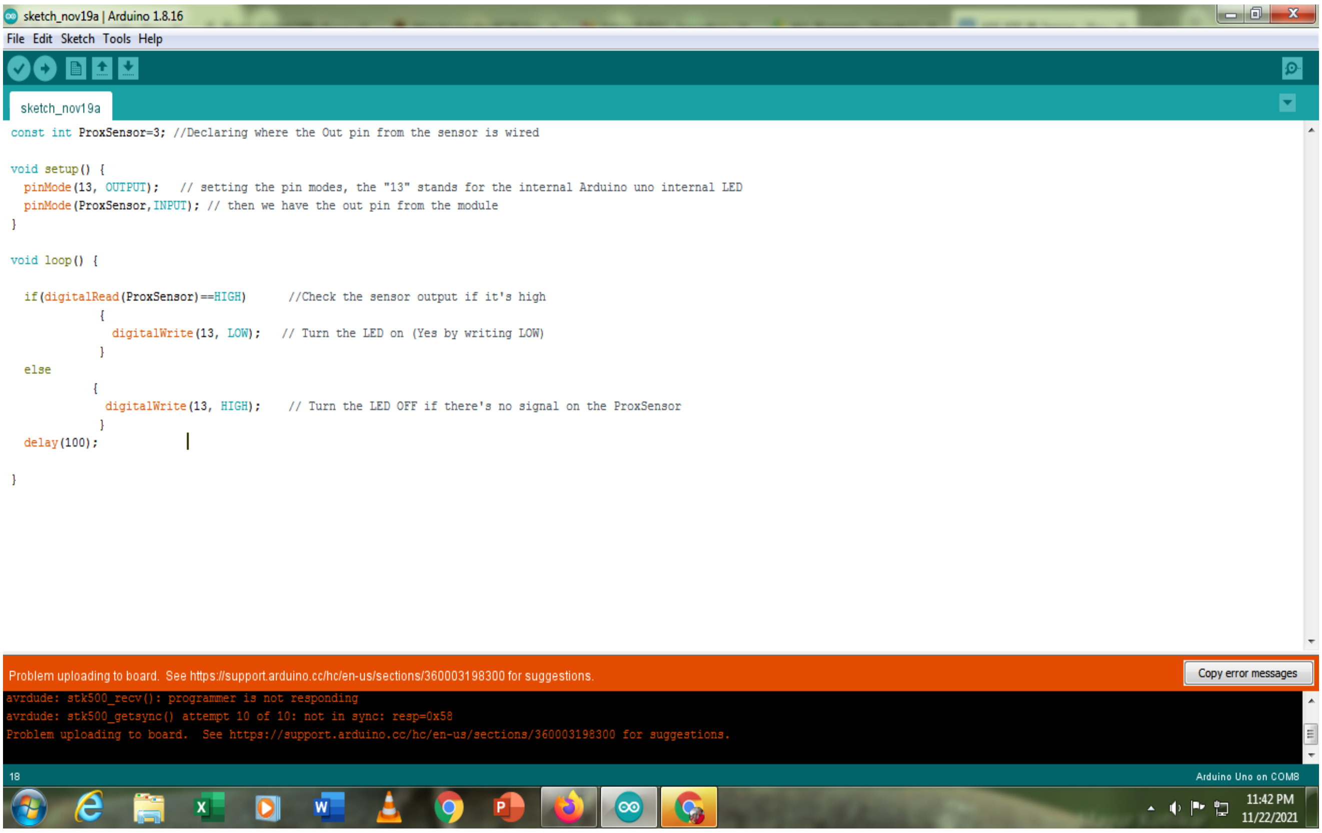
Signal LED – Illuminates when obstacle is detected

Variable resistor – Adjust detection distance. CCW decreases distance. CW increases distance.

Source Code

```
 void setup() {
  // put your setup code here, to run once:
pinMode(2,INPUT);
pinMode(13,OUTPUT);//LED
}
void loop() {
  // put your main code here, to run repeatedly:
```

```
if(digitalRead(2)==LOW){
  digitalWrite(13,HIGH);
}
else{
  digitalWrite(13,LOW);
}
}
```



```
const int ProxSensor=3; //Declaring where the Out pin from the sensor is wired

void setup() {
  pinMode(13, OUTPUT);   // setting the pin modes, the "13" stands for the internal Arduino uno internal LED
  pinMode(ProxSensor,INPUT); // then we have the out pin from the module
}

void loop() {

  if(digitalRead(ProxSensor)==HIGH)      //Check the sensor output if it's high
             {
              digitalWrite(13, LOW);   // Turn the LED on (Yes by writing LOW)
             }
  else
            {
             digitalWrite(13, HIGH);    // Turn the LED OFF if there's no signal on the ProxSensor
             }
  delay(100);

}
```

Problem uploading to board. See https://support.arduino.cc/hc/en-us/sections/360003198300 for suggestions.

```
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x58
Problem uploading to board.  See https://support.arduino.cc/hc/en-us/sections/360003198300 for suggestions.
```

Conclussion : Succfully implemented Understanding the connectivity of Raspberry-Pi / Adriano with IR sensor

Assignment No. 07

Problem Statement: Understanding the connectivity of Raspberry-Pi /Beagle/ Arduino UNO board circuit with temperature sensor. Write an application to read the environment temperature. If temperature crosses a threshold value, generate alerts using LEDs.

Objectives:

To Learn the interfacing of LM35, with Arduino UNO

Outcome:

After completion of this assignment students will be able to understand the use of LM35 with the Arduino UNO and use of functions like void setup() and void loop (),Serial.println().

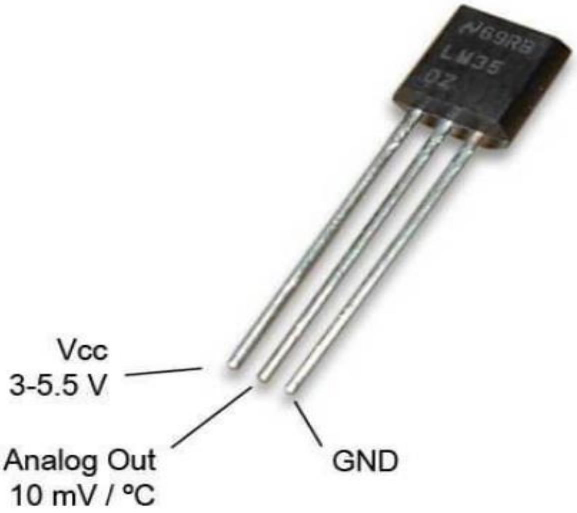
Software & Hardware Requirements:

1. Operating System: Windows (XP/Vista/7/10)
2. Software: Arduino IDE 1.8.3
3. Hardware: Raspberry -pi,,Arduino UNO, LM35,Patch Cords, USB cable type A/B.

Theory:

The Temperature Sensor LM35 series are precision integrated-circuit temperature devices with an output voltage linearly proportional to the Centigrade temperature.

The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full −55°C to 150°C temperature range.



Technical Specifications

- Calibrated directly in Celsius (Centigrade)
- Linear + 10-mV/°C scale factor
- 0.5°C ensured accuracy (at 25°C)
- Rated for full −55°C to 150°C range
- Suitable for remote applications

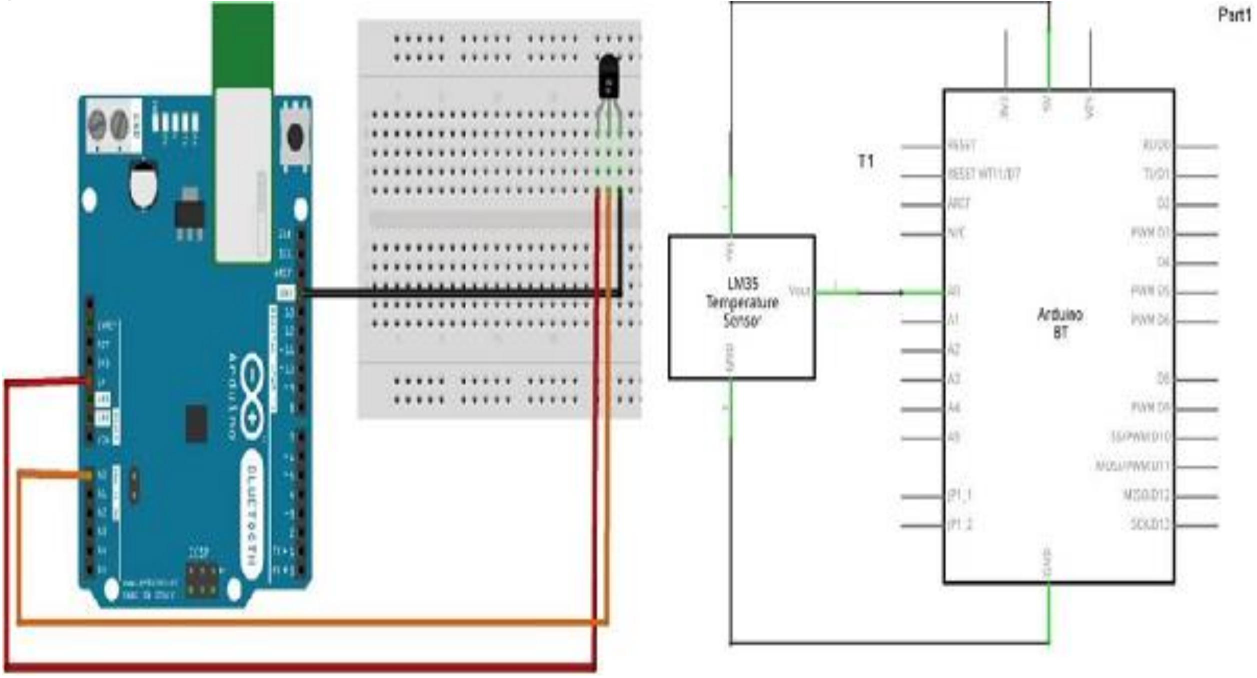
Components Required

You will need the following components −

- 1 × Breadboard
- 1 × Arduino Uno R3
- 1 × LM35 sensor

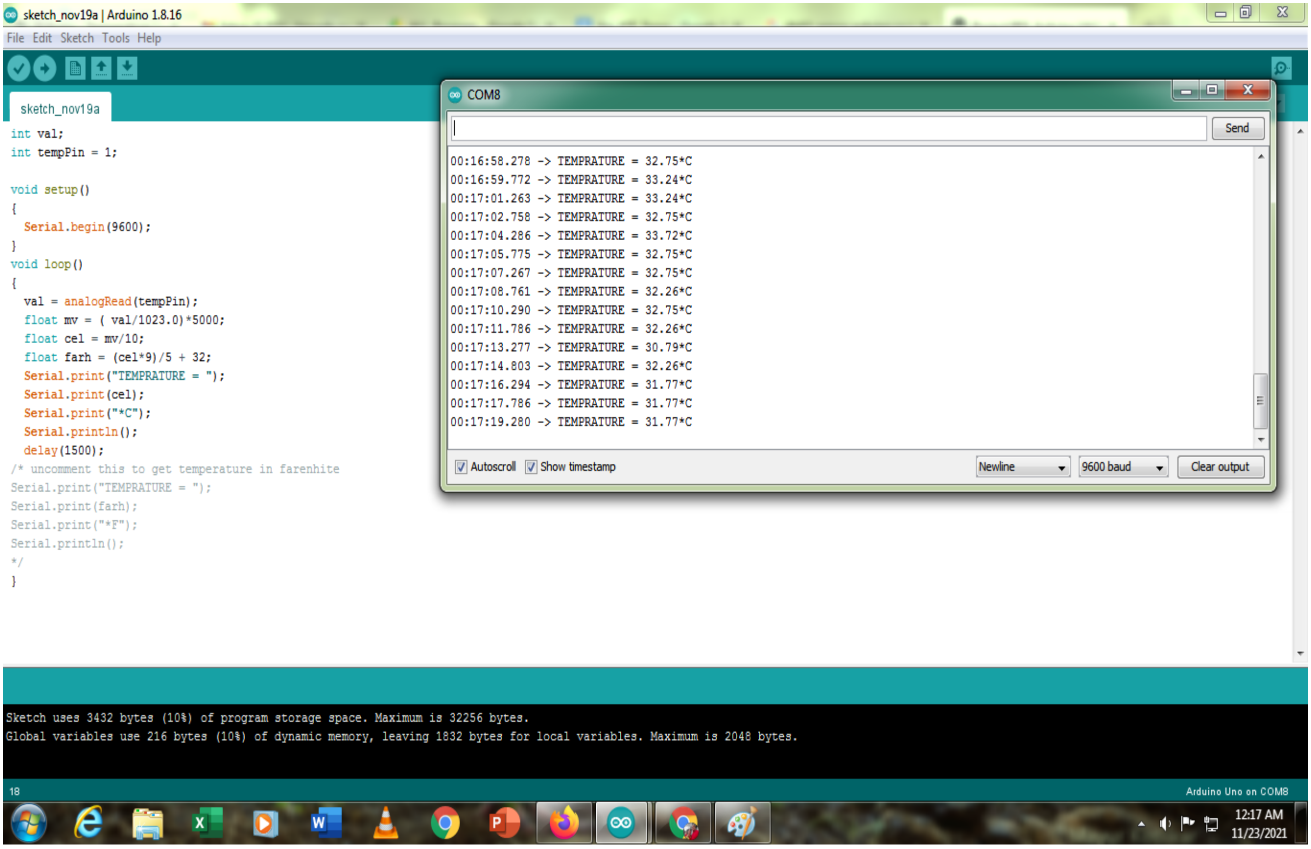
Procedure

Follow the circuit diagram and hook up the components on the breadboard as shown in the image given below.



Source Code:

```
int tempPin = 1;
void setup()
{
  Serial.begin(9600);
}
void loop()
{
  val = analogRead(tempPin);
  float mv = ( val/1023.0)*5000;
  float cel = mv/10;
  float farh = (cel*9)/5 + 32;
  Serial.print("TEMPRATURE = ");
  Serial.print(cel);
  Serial.print("*C");
  Serial.println();
  delay(1500);
/* uncomment this to get temperature in farenhite
Serial.print("TEMPRATURE = ");
Serial.print(farh);
Serial.print("*F");
Serial.println();
*/
}
```



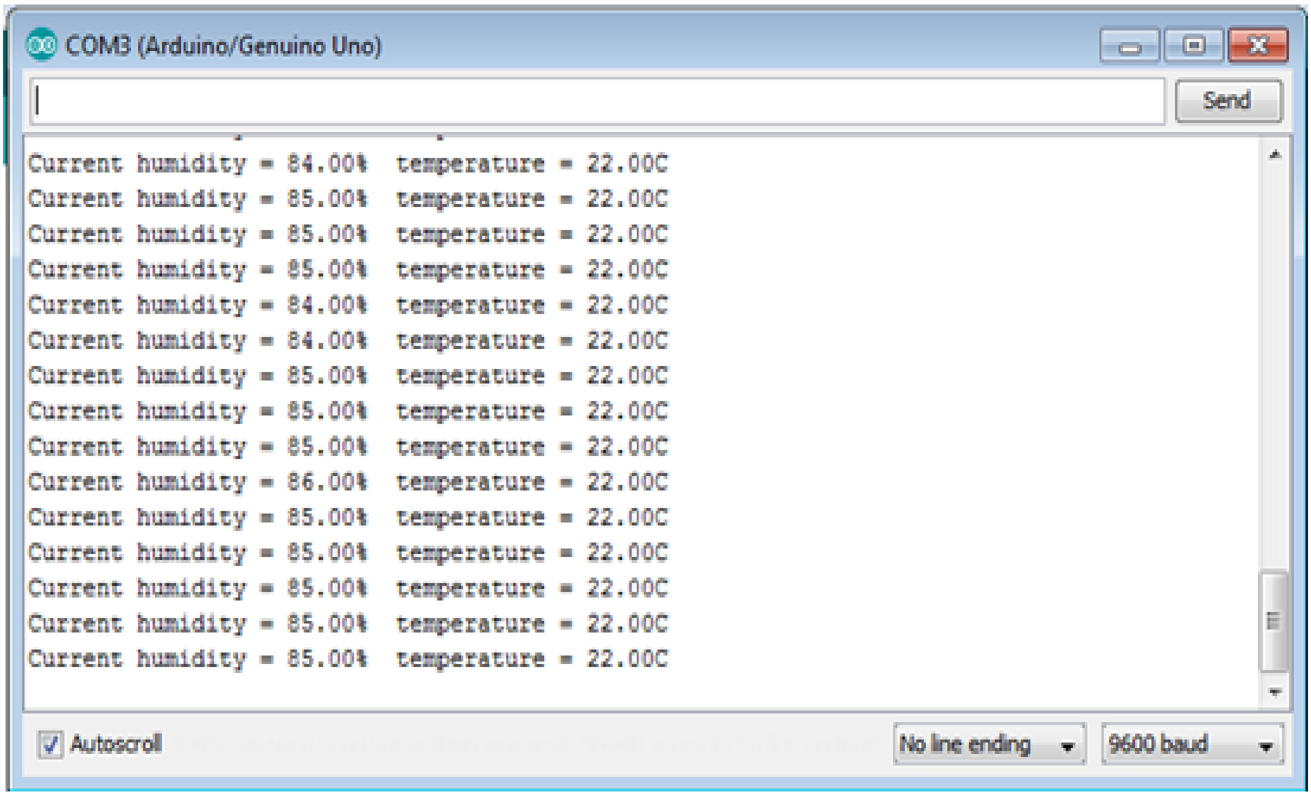
```
int val;
int tempPin = 1;

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  val = analogRead(tempPin);
  float mv = ( val/1023.0)*5000;
  float cel = mv/10;
  float farh = (cel*9)/5 + 32;
  Serial.print("TEMPRATURE = ");
  Serial.print(cel);
  Serial.print("*C");
  Serial.println();
  delay(1500);
/* uncomment this to get temperature in farenhite
Serial.print("TEMPRATURE = ");
Serial.print(farh);
Serial.print("*F");
Serial.println();
*/
}
```

```
00:16:58.278 -> TEMPRATURE = 32.75*C
00:16:59.772 -> TEMPRATURE = 33.24*C
00:17:01.263 -> TEMPRATURE = 33.24*C
00:17:02.758 -> TEMPRATURE = 32.75*C
00:17:04.286 -> TEMPRATURE = 33.72*C
00:17:05.775 -> TEMPRATURE = 32.75*C
00:17:07.267 -> TEMPRATURE = 32.75*C
00:17:08.761 -> TEMPRATURE = 32.26*C
00:17:10.290 -> TEMPRATURE = 32.75*C
00:17:11.786 -> TEMPRATURE = 32.26*C
00:17:13.277 -> TEMPRATURE = 30.79*C
00:17:14.803 -> TEMPRATURE = 32.26*C
00:17:16.294 -> TEMPRATURE = 31.77*C
00:17:17.786 -> TEMPRATURE = 31.77*C
00:17:19.280 -> TEMPRATURE = 31.77*C
```

☑ Autoscroll ☑ Show timestamp Newline 9600 baud

```
Sketch uses 3432 bytes (10%) of program storage space. Maximum is 32256 bytes.
Global variables use 216 bytes (10%) of dynamic memory, leaving 1832 bytes for local variables. Maximum is 2048 bytes.
```

Output:



```
Current humidity = 84.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 84.00%  temperature = 22.00C
Current humidity = 84.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 86.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
Current humidity = 85.00%  temperature = 22.00C
```

☑ Autoscroll   No line ending   9600 baud

Conclusion : Successfully understood the connectivity of the Arduino UNO circuit with temperature sensor.

Part II: Elective I

Software Project Management
Assignment No. 08

Software project Management: implement one mini project (hotel management, HR management, Bank management etc or perform any software mini projects)

Problem Statement: Perform one mini project & Create Project Plan

▪ Specify project name and start (or finish) date.
▪ Identify and define project tasks.
▪ Define duration for each project task.
▪ Define milestones in the plan
▪ Define dependency between tasks
▪ Define project calendar.
▪ Define project resources and specify resource type
▪ Assign resources against each task and baseline the project plan

Objectives:

3. To learn software project management
4. To identify project tasks &define project calendar

Theory:

Note: Implement one mini project & write theory observations**

- Specify project name and start (or finish) date.
- Identify and define project tasks.
- Define duration for each project task.
- Define milestones in the plan
- Define dependency between tasks
- Define project calendar.
- Define project resources and specify resource type
- Assign resources against each task and baseline the project plan
- We have created Project Plan.
- Identify and define project tasks.

Conclusion:

Successfully defined project plan with milestones & project resource

Assignment No. 09

Software project Management: implement one mini project (hotel management, HR management, Bank management etc or perform any software mini projects)

Problem Statement: Perform one mini project &Execute and Monitor Project Plan

- Update % Complete with current task status.
- Review the status of each task.
- Compare Planned vs Actual Status
- Review the status of Critical Path
- Review resources assignation status

Objectives:

5. To learn software project management by monitoring
6. To review the project status & compare both

Theory:
Note**: Implement one mini project & write theory observations Perform one mini project &**

Execute and Monitor Project Plan

- Update % Complete with current task status.
- Review the status of each task.
- Compare Planned vs Actual Status
- Review the status of Critical Path
- Review resources assignation status
- To learn how to calculate project completion.
- Review the tasks status

Conclusion:

Successfully defined project plan with milestones & project resource