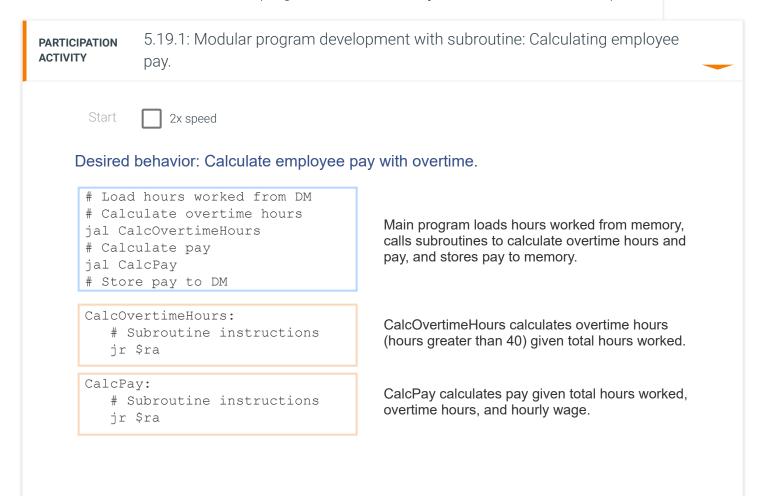
5.19 Assembly program example: Subroutines

Assembly program

Subroutines enable programmers to write modular assembly programs. A subroutine has well-defined input and output, so can focus on developing a particular subroutine (or module) independently of other subroutines. Each subroutine should he recognizable behavior, and the main behavior of the program should be easily understandable via a sequence of subroutine



PARTICIPATION ACTIVITY	5.19.2: Modular program development.	_
Refer to the ar	nimation above.	
program in subroutines tested sepa	evelopment means to divide a to separate modules (or s) that can be developed and arately.	
O True O False		
consists of subroutines O True		•
	ay subroutine can be written CalcOvertimeHours	

Modular subroutine development

The subroutines for calculating the number of overtime hours and calculating the employee's pay can be developed separa

Overtime is the number of hours worked beyond 40 hours in a single week. Ex: If an employee works 55 hours, the employe hours of overtime. If an employee works 40 hours or fewer, then the employee worked zero overtime hours. The CalcOvert

subroutine below calculates the number of overtime hours an employee has worked given the number of total hours the er in a week. \$t0 is used for the subroutine's argument, which is the number of hours worked in a week. \$t1 is used for the sul value, which is the number of overtime hours.

Figure 5.19.1: CalcOvertimeHours subroutine calculates an employee's overtime hours.

```
CalcOvertimeHours:
   addi $t2, $zero, 40
   slt $t3, $t0, $t2
   bne $t3, $zero, NoOvertime
   # Overtime worked
   # Overtime hours is 40 - hours worked
   sub $t1, $t0, $t2
   j ReturnOvertime
NoOvertime:
   # No overtime, so overtime hours is 0
   addi $t1, $zero, 0
ReturnOvertime:
   jr $ra
```

PARTICIPATION ACTIVITY 5.19.3: CalcOvertimeHours. 1) If \$t0 holds 35, what is \$t1 after the CalcOvertimeHours subroutines returns? 0 0 0 35 0 40 2) If \$t0 holds 42, what is \$t1 after the CalcOvertimeHours subroutines returns? 0 0

Q 2
Q 40
3) If the employee did not work overtime, which instruction writes 0 to the register for the return value?
Q addi \$t2, \$zero, 40
Q addi \$t1, \$zero, 0
Q sub \$t1, \$t0, \$t2

An employee is paid an hourly wage for the first 40 hours, and two times the hourly wage for overtime hours. The CalcPay calculates an employee's weekly pay. An employee's hourly pay rate is \$10/hour. The subroutine's uses \$t0 for the total hour for the employee hourly wage, and \$t2 for the number of overtime hours. The subroutine returns the employee's pay using

Figure 5.19.2: CalcPay subroutine calculates an employee's pay.

CalcPay:
 # Calculate base pay
 mul \$t3, \$t0, \$t1
 # Calculate overtime pay
 mul \$t4, \$t2, \$t1
 # Calculate total pay
 add \$t3, \$t3, \$t4
 jr \$ra

```
PARTICIPATION ACTIVITY

5.19.4: CalcPay subroutine.

1) Which register is used for the hourly wage?

O $t1
```

	O \$t2
ł	The total pay is calculated as the total nours times the hourly wage plus the overtime hours times the hourly wage.
	O True
	O False
ŀ	f \$t0 holds 30, \$t1 holds 10, and \$t2 holds 0, what is \$t3 after the subroutine returns? O 300 O 600
ŀ	f \$t0 holds 55, \$t1 holds 10, and \$t2 holds 15, what is \$t3 after the subroutine returns? O 550 O 700

Main program behavior is a sequence of subroutine calls.

The program below calculates the pay for a single employee, where DM[5000] is the total hours worked by the employee, a is stored to DM[5040]. The program's main behavior consists of loading the hours worked, calling the CalcOvertimeHours s calculate the overtime hours, calling the CalcPay subroutine to calculate the pay, and storing the pay to memory.

Figure 5.19.3: Calculating pay for a single employee.

```
# Load hours worked from DM[5000]
addi $t6, $zero, 5000
lw $t0, 0($t6)
jal CalcOvertimeHours
# Overtime hours returned in $t1
# Copy $t1 to $t2
add $t2, $zero, $t1
# Initialize pay rate to $10/hour
addi $t1, $zero, 10
jal CalcPay
# Pay is returned in $t3
# Store pay to DM[5040]
addi $t6, $zero, 5040
sw $t3, 0($t6)
j Done
CalcOvertimeHours:
  addi $t2, $zero, 40
  slt $t3, $t0, $t2
  bne $t3, $zero, NoOvertime
  # Overtime worked
  # Overtime hours is 40 - hours worked
  sub $t1, $t0, $t2
  j ReturnOvertime
NoOvertime:
  # No overtime, so overtime hours is 0
  addi $t1, $zero, 0
ReturnOvertime:
  jr $ra
CalcPay:
  # Calculate base pay
  mul $t3, $t0, $t1
  # Calculate overtime pay
  mul $t4, $t2, $t1
  # Calculate total pay
  add $t3, $t3, $t4
  jr $ra
Done:
```

PARTICIPATION ACTIVITY

5.19.5: Using the subroutines to calculate employee pay.

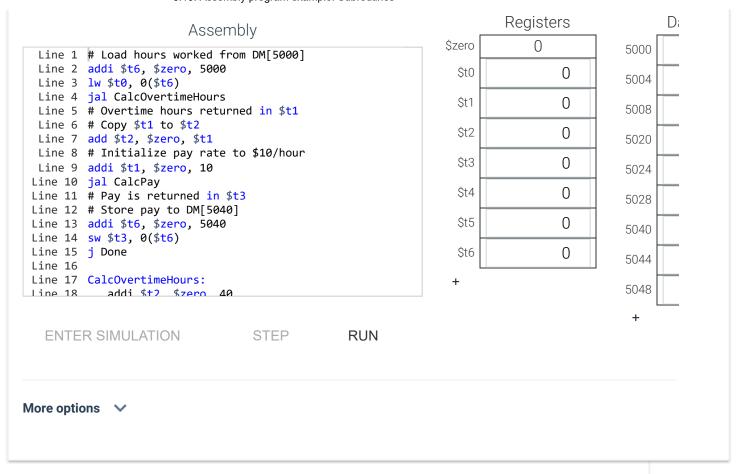
Refer to the program above. 1) What does add \$t2, \$zero, \$t1 do? O Initializes the total pay. O Copies \$t1 to \$t2. 2) Which instruction writes the total hours worked argument for CalcPay? O addi \$t1, \$zero, 10 O add \$t2, \$zero, \$t1 O lw \$t0, 0(\$t6) 3) What does j Done do? O Calls the done subroutine. O Jumps past the CalcOvertimeHour and CalcPay subroutines.

PARTICIPATION ACTIVITY

5.19.6: Calculating pay for multiple employees.

Extend the program below to calculate pay for multiple employees, where each employee has a different hourly wage.

- 1. Modify the program to calculate pay for three employees. DM[5000], DM[5004], and DM[5008] are the total hours worked for the three employees. Store the employees' pay to DM[5040], DM[5044], and DM[5048], respectively.
- 2. Modify the program to load the employees' pay rates from DM[5020], DM[5024], and DM[5028], respectively.



Provide feedback on this section