

```
Import JJIO
```

```
Class BigPay
```

```
-- Name An Onmymous  
-- Does pay as one big monolith; no sub-programs
```

```
Constant Break1 = 40 -- overtime pay limit  
Constant Break2 = 60 -- double time limit
```

```
Routine NetPays (none) is private  
  
Boxes hours, gross, net, misc ofType int  
Boxes taxRate, taxes, deductions ofType int
```

```
-- Does net pay repeatedly
```

```
Start
```

```
Repeat
```

```
-- Input of data  
Output "Enter hours worked "  
Input hours  
Outputln hours
```

```
Repeat -- on error input  
ExitOn (hours <= 7 * 24)  
Output "Error.. ReEnter "  
Input hours  
Outputln hours  
EndRepeat
```

```
-- Other inputs (rate = 10)
```

```
ExitOn (hours < 0)
```

```
-- Gross Pay
```

```
If (hours > 60) then -- extended  
Set gross = 700 + 2*10*(hours - 60)  
Else  
If (hours < 40) then -- original  
Set gross = 10 * hours  
Else  
Set gross = 40*10 + 15*(hours - 40)  
EndIf -- original part  
EndIf -- extended part
```

```
-- Deductions
```

```
Set taxRate = 20 -- percent could input  
Set taxes = gross * taxRate / 100 --!!  
Output "Enter misc deductions "  
Input misc  
Outputln misc  
Set deductions = taxes + misc
```

```
Set net = gross - deductions
```

```
-- Output
```

```
Output "The gross pay is "  
Outputln gross
```

```
Output "The net pay is "  
Outputln net  
Outputln " "
```

```
EndRepeat
```

```
EndRoutine NetPays
```

```
EndClass BigPay
```

```
Import JJIO
```

```
Class BigPay
```

```
-- Name A Nonymous  
-- Does pay in parts; using sub-programs
```

```
Constant Break1 = 40 -- overtime pay limit  
Constant Break2 = 60 -- double time limit
```

```
Routine NetPays (none) is private  
Boxes hours, gross, net ofType int  
-- Does net pay repeatedly  
Start  
Repeat  
-- Input of data  
Output "Enter hours worked "  
Input hours  
Outputln hours  
Repeat -- on error input  
ExitOn (hours <= 7 * 24)  
Output "Error.. ReEnter "  
Input hours  
Outputln hours  
EndRepeat  
ExitOn (hours < 0)  
Output "The gross pay is "  
Outputln grossPay (hours)  
Set net = gross - deductions (gross)  
Output "The net pay is "  
Outputln net  
Outputln " "  
EndRepeat  
EndRoutine NetPays -- main
```

```
Function grossPay (hours) ofType int is public  
Slot hours ofType int  
Box result ofType int  
Box gross ofType int  
-- Does compute and return gross pay  
If (hours > 60) then -- extended  
Set gross = 700 + 2*10 * (hours - 60)  
Else  
If (hours < 40) then -- original  
Set gross = 10 * hours  
Else  
Set gross = 40*10 + 15 * (hours - 40)  
EndIf -- original part  
EndIf -- extended part  
Set result = gross  
EndFunction grossPay
```

```
Function deductions (gross) ofType int is public  
Slot gross ofType int  
Box result ofType int  
Boxes taxes, taxRate, misc ofType int  
-- Does compute deductions  
Set taxRate = 20 -- percent could input  
Set taxes = gross * taxRate / 100 --!!  
Output "Enter misc deductions "  
Input misc  
Outputln misc  
Set result = taxes + misc  
EndFunction deductions
```

```
EndClass BigPay
```