Programs done on Nov 4th:

GPA 2:

With a check for whether it is a valid grade

```
#How to fix this zero div error?
points = {'A':10.0, 'B':8.0, 'C':6.0, 'D':4.0,'F':0.0}

num_courses =0
total_points= 0
done = False

while not done:
    grade = input()
    if grade == '':
        done=True
    elif grade not in points:
        print("grade ignored",grade)
    else:
        total_points += points[grade]
        num_courses=num_courses+1


print("Your GPA is: ", total_points/num_courses)
```

GPA with a check on number of courses:

```
points = {'A':10.0, 'B':8.0, 'C':6.0, 'D':4.0,'F':0.0}

num_courses =0
total_points= 0
done = False

while not done:
    grade = input()
    if grade == '':
        done=True
    elif grade not in points:
        print("grade ignored",grade)
    else:
        total_points += points[grade]
        num_courses=num_courses+1

if num_courses>0:

  print("Your GPA is: ", total_points/num_courses)
```

GPA implemented as a function (and with a default parameter):

```
def compute_gpa(grades, points = {'A':10.0, 'B':8.0,'C':6.0,'D':4.0,'F':0.0}):
# 2nd argument is called a default arg

num_courses=0
total_points=0

for g in grades:
    if g in points:

        num_courses=num_courses+1
        total_points = total_points+points[g]
       # in the notation points[g], Python accesses the dict obj with the key value indicated by  g
return total_points / num_courses
```

Program to check a valid identifier:

```
#Initializations
lower_case_alphabet = 'abcdefghijklmnopqrstuvwxyz'
upper_case_alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
under_score = '_'

digit = '0123456789'

# Function valid_First_Char(string) returns True if first char in identifier is a letter else returns False
def valid_First_Char(string):
    return (string[0] in lower_case_alphabet) or (string[0] in upper_case_alphabet) or (string[0]=='_')

#Function valid_Non_First_Char(kar) returns True if a character from the 2nd position onwards is an upper or lower case
#letter or an underscore character or a character representing a single digit
def valid_Non_First_Char(kar):

    return (kar in lower_case_alphabet) or (kar in upper_case_alphabet) or (kar in digit) or (kar == under_score)#Short-circuit evaluation


#The function iS_It_A_Valid_Identifier(string) returns True if string is a valid identifier;
#else returns False
#The function assumes that string is a valid identifier initially (invalid_Identifier=False)
#if any of the rules for string to be a valid identifier is found to be violated, the function
#returns False
def iS_It_A_Valid_Python_Identifier(string):
```

```
#invalid_Identifier=False

if valid_First_Char(string):

   i=1
   while  (i<len(string)): # 4 < 4  which is False and the loop terminates at this stage
      if valid_Non_First_Char(string[i]):
         i=i+1
      else:
         break;
   return (i==len(string))        # 3 == 4  which evaluates to False
else:
   return False

return (i==len(string))
```