Samples **Programming with Python SOLUTIONS**

1. The full Python inputs and results are shown below. Please note that there may be some slight rounding errors present, and that the results may not be formatted exactly as you would find in Python.

Here is the program:

```
from __future__ import division
from pylab import *

def newtonGravity(m1, m2, d):
    # Newton's Law of Universal Gravitation
    G = 6.67*10.0**(-11)
    F = G*m1*m2/d**2
    return F

def gravityX(mass, distance):
    # Gravitational force for planet X
    massX = 8.0*10.0**27
    return newtonGravity(massX, mass, distance)

# Information about planet X
radius = 7863*10.0**3

# Marcus' mass (in kg) and height above the planet (in m)
mass = 65
height = 66702*10.0**3

# Calculate the distance from Marcus to the centre of the planet
dist = height+radius

# Calculation the gravitational force on Marcus
F = gravityX(mass, dist)
print "Marcus is experiencing a gravitational force of", F, "N."

# Calculating the gravitational acceleration on the surface
# We can use our function by using a "mass" of 1kg
g = gravityX(1, radius)
print "The gravitational acceleration on the surface is", g, "m/s^2."
```

Here is the output from running the program:

```
Marcus is experiencing a gravitational force of 6238.19768 N.
The gravitational acceleration on the surface is 8630.56583 m/s^2.
```

2. The full Python inputs and results are shown below. Please note that there may be some slight rounding errors present, and that the results may not be formatted exactly as you would find in Python.

Here is the program:

```
from __future__ import division
from pylab import *

def newtonGravity(m1, m2, d):
    # Newton's Law of Universal Gravitation
    G = 6.67*10.0**(-11)
```

```
    F = G*m1*m2/d**2
    return F

def gravityX(mass, distance):
    # Gravitational force for planet X
    massX = 4.4*10.0**28
    return newtonGravity(massX, mass, distance)

# Information about planet X
radius = 6297*10.0**3

# Marcus' mass (in kg) and height above the planet (in m)
mass = 66
height = 53175*10.0**3

# Calculate the distance from Marcus to the centre of the planet
dist = height+radius

# Calculation the gravitational force on Marcus
F = gravityX(mass, dist)
print "Marcus is experiencing a gravitational force of", F, "N."

# Calculating the gravitational acceleration on the surface
# We can use our function by using a "mass" of 1kg
g = gravityX(1, radius)
print "The gravitational acceleration on the surface is", g, "m/s^2."
```

Here is the output from running the program:

```
Marcus is experiencing a gravitational force of 54764.2770 N.
The gravitational acceleration on the surface is 74013.5310 m/s^2.
```

3. The full Python inputs and results are shown below. Please note that there may be some slight rounding errors present, and that the results may not be formatted exactly as you would find in Python.

Here is the program:

```
from __future__ import division
from pylab import *

def newtonGravity(m1, m2, d):
    # Newton's Law of Universal Gravitation
    G = 6.67*10.0**(-11)
    F = G*m1*m2/d**2
    return F

def gravityX(mass, distance):
    # Gravitational force for planet X
    massX = 9.0*10.0**25
    return newtonGravity(massX, mass, distance)

# Information about planet X
radius = 6005*10.0**3

# Marcus' mass (in kg) and height above the planet (in m)
mass = 77
```

```
height = 41287*10.0**3

# Calculate the distance from Marcus to the centre of the planet
dist = height+radius

# Calculation the gravitational force on Marcus
F = gravityX(mass, dist)
print "Marcus is experiencing a gravitational force of", F, "N."

# Calculating the gravitational acceleration on the surface
# We can use our function by using a "mass" of 1kg
g = gravityX(1, radius)
print "The gravitational acceleration on the surface is", g, "m/s^2."
```

Here is the output from running the program:

```
Marcus is experiencing a gravitational force of 206.672982 N.
The gravitational acceleration on the surface is 166.472430 m/s^2.
```