HELENE-WEBER-SCHULE BUCHEN	Grundlagen der Programmierung **Probe-Klassenarbeit**	**Vorname:** **Nachname:** **Datum:**

Maximale Punktzahl: **10**, davon erreicht: _____ **Note:** _____

Theorieteil

Aufgabe 1: Multiple Choice Fragen (5 Punkte)

Kreuzen Sie die richtigen Antworten zu den folgenden Fragen an.

a) Welche der folgenden Aussagen beschreibt eine gültige Variable in Python?

- ○ `1zahl = 5`
- ○ `boolean-1 = True`
- ○ `mein_variable = "Hallo Welt!"`
- ○ `if = 10`

b) Welcher Operator wird in Python verwendet, um eine Division durchzuführen?

- ○ `/`
- ○ `//`
- ○ `%`
- ○ `|`

c) Welche Bedingung wird durch den folgenden Python-Code überprüft?

```
if x >= 10:
    print("x")
```

- ○ X ist größer als 10
- ○ x ist kleiner als 10
- ○ x ist größer gleich 10
- ○ x ist kleiner gleich 10

d) Welches Schlüsselwort wird bei einer Fallunterscheidung verwendet?

- ○ `while`
- ○ `for`
- ○ `if`
- ○ `when`

e) Welches Ergebnis wird mit der folgenden Anweisung in Python ausgegeben?

```
print(5 * 3)
```

- ○ 15
- ○ 53
- ○ 5*3
- ○ Fehler, weil * kein gültiger Operator ist.

HELENE-WEBER-SCHULE BUCHEN	Grundlagen der Programmierung **Probe-Klassenarbeit**	**Vorname:** **Nachname:** **Datum:**

Aufgabe 2: Grundlagen (10 Punkte)

1. **Nennen Sie <u>zwei</u> Schlüsselwörter für Fallunterscheidungen** (1 Punkt)

   if , else

2. **Nennen Sie die Datentypen der folgenden Variablen** (2 Punkte)

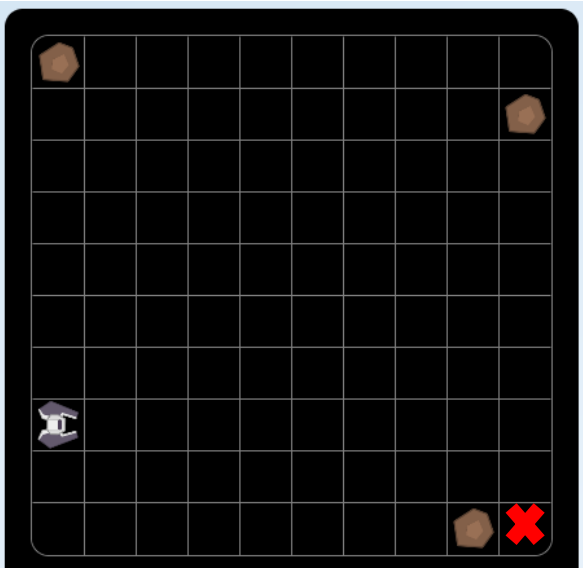
variable_1 = "Informatik" String

variable_2 = 777 Integer

variable_3 = "5.55" String

3. **Kreuzen Sie das Feld an, auf dem die Argo nach dem Programmdurchlauf stehen bleibt.** (3 Punkte)

```
1  turnLeft()
2  while not rockFront():
3    move()
4  turnRight()
5  while not rockFront():
6    move()
7  turnRight()
8  while not rockFront():
9    move()
10 turnLeft()
11 move()
12 turnRight()
13 move()
```



4. **Erklären das Konzept der Schleife und nennen Sie ein Vorteil** (3 Punkte)

Eine Schleife ist ein Programmierkonstrukt, dass Anweisungssequenzen wiederholt ausführt, solange eine bestimmte Bedingung erfüllt ist.

Vorteile: Reduzierung von Programmcode

HELENE-WEBER-SCHULE BUCHEN	Grundlagen der Programmierung **Probe-Klassenarbeit**	**Vorname:** **Nachname:** **Datum:**

Anwendungsteil

Aufgabe 3: Steuerung der Argo (20 Punkte, je Teilaufgabe 10 Punkte)

Für die Steuerung der Argo stehen folgende Anweisungen zur Verfügung:

Anweisung	Aktion
move()	Die Argo fliegt einen Sektor nach vorne.
turnRight()	Die Argo dreht sich um 90° im Uhrzeigersinn
turnLeft()	Die Argo dreht sich um 90° gegen den Uhrzeigersinn
putPow()	Die Argo legt ein PowerUp in den aktuellen Sektor
removePow()	Die Argo entfernt ein PowerUp aus dem aktuellen Sektor
rockFront()	vor einem Asteroiden?
rockLeft()	links ein Asteroid?
rockRight()	rechts ein Asteroid?
onPow()	auf einem PowerUp?

a) Die Argo soll bis zum Asteroid fliegen und dabei folgendermaßen arbeiten.
   1. Ist rechts oder links von der Argo ein Asteroid, dann soll ein PowerUp auf den Weg gelegt werden.
   2. Wenn kein Asteroid rechts oder links neben der Argo ist, dann soll nichts gemacht werden.
   3. Ist nur rechts oder links ein Asteroid dann soll auf dem weg und in die Lücke jeweils ein PowerUp gelegt werden.

   Schreibe ein Programm, damit Argo diese Aufgabe lösen kann.
   Hinweis: Der Programmcode soll auch für unterschiedlich lange Reihen funktionieren.

1. Öffnen Sie auf https://inf-schule.de/tools/spacebug die Datei *Aufgabe3a.json*
2. Erstellen Sie das Programm wie in der Aufgabenstellung vorgegeben.
3. Speichern Sie Ihren Programmcode unter "Lösungscode" ab.

Lösungscode

```
while not rockFront():
  move()
  if rockLeft() and rockRight():
    putPow()

  if rockLeft() and not rockRight():
    putPow()
    turnRight()
    move()
    putPow()
    turnLeft()
    turnLeft()
    move()
    turnRight()

  if rockRight() and not rockLeft():
    putPow()
    turnLeft()
    move()
    putPow()
    turnRight()
    turnRight()
    move()
    turnLeft()
```

b) Die Argo steht vor einer Asteroidenreihe. Schreibe ein Programm, damit Argo um die Reihe herumfliegt und auf der anderen Seite in die Mitte hält.

Hinweis: Der Programmcode soll auch für unterschiedlich lange Reihen funktionieren.

1. Öffnen Sie auf https://inf-schule.de/tools/spacebug die Datei *Aufgabe3b.json*
2. Erstellen Sie das Programm wie in der Aufgabenstellung vorgegeben.
3. Speichern Sie Ihren Programmcode unter "Lösungscode" ab.

Lösungscode

```
turnLeft()
move()
turnRight()
move()
i = 0

#Ab hier zählen
while rockRight():
  i = i+1
  move()

#Die Argo drehen
turnRight()
move()
move()
turnRight()

#Ab hier die halbe Strecke zurück liegen
halbeStrecke = i/2
while halbeStrecke > 0:
  halbeStrecke = halbeStrecke-1
  move()
```

Aufgabe 4: Eigene Programme erstellen (15 Punkte)

a) **Begrüßung:** Schreibe ein Python-Programm, das den Benutzer nach seinem Namen fragt und ihn dann begrüßt. (2 Punkte)

```
Kommandozeile
>>> %Run -c $EDITOR_CONTENT
 Gebe deinen Namen ein: Max
 Hallo Max
```

```
name = input("Gebe deinen Namen ein: ")
print("Hallo", name)
```

b) **Addiere zwei Zahlen:** Erstelle ein Programm, das den Benutzer nach zwei Zahlen fragt und die Summe dieser Zahlen berechnet und ausgibt. (4 Punkte)

```
Kommandozeile
>>> %Run -c $EDITOR_CONTENT
 Gebe die erste Zahl ein: 5
 Gebe die zweite Zahl ein: 3
 8
```

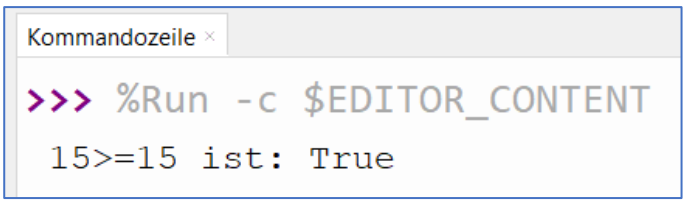
```
zahl1 = int(input("Gebe die erste Zahl ein: "))
zahl2 = int(input("Gebe die zweite Zahl ein: "))
print(zahl1+zahl2)
```

c) **Stunden in Minuten umrechnen:** Schreibe ein Programm, das den Benutzer nach einer Anzahl von Stunden fragt und die Anzahl der Minuten berechnet. Anschließend soll das Ergebnis mit der Nachricht: ***… Stunden sind …. Minuten*** (4 Punkte)

```
Kommandozeile
>>> %Run -c $EDITOR_CONTENT
 Gebe die Stunden ein, die in Minuten umgerechnet werden sollen: 2
 2 Stunden sind 120 Minuten
```

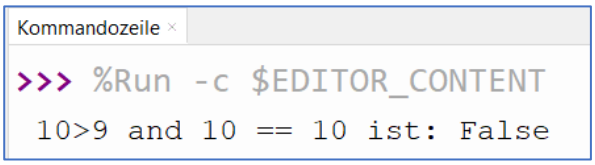
```
zahl1 = int(input("Gebe die Stunden ein, die in Minuten umgerechnet
werden sollen: "))
print(zahl1, "Stunden sind", zahl1*60, "Minuten")
```

d) Vergleichen: Schreibe ein Programm, das überprüft, ob 15 größer oder gleich 15 ist. (2 Punkte)




```
print("15>=15 ist:", 15>=15)
```

e) `Vergleichen:` Schreibe ein Programm, das überprüft, ob 8 größer als 9 und gleich 8 ist. (3 Punkte)




```
print("10>9 and 10 == 10 ist:", 8>9 and 8 == 8)
```

Bewertungsskala / Notenschlüssel

Punkte	Note
48 - 50	15
45 - 47	14
43 - 44	13
40 - 42	12
38 - 39	11
35 - 37	10
33 - 34	9
30 - 32	8
28 - 29	7
25 - 27	6
23 - 24	5
20 - 22	4
17 - 19	3
14 - 16	2

Punkte	Note
10 - 13	1
0 - 9	0