G.C.E. O/L ICT · 11 ශ්‍රේණිය · පාඩම 1

1.10 උප වැඩසටහන්: ශ්‍රිත සහ ක්‍රියා පටිපාටි

විෂය නිර්දේශය 10.10. උප වැඩසටහනක් යනු **එක් කාර්යයක්** කරන, නමක් ඇති වැඩසටහනක කුඩා කොටසකි. අප එය එක් වරක් ලියා අවශ්‍ය සෑම විටම **කැඳවමු** (භාවිත කරමු) – නැවත නොලියා දිනපතා අනුගමනය කළ හැකි පොල් සම්බෝල වට්ටෝරුවක් මෙන්. විශාල වැඩසටහන් කුඩා උප වැඩසටහන් බොහොමයකින් සාදා ඇත.

1. උප වැඩසටහන් භාවිත කරන්නේ ඇයි?

නැවත භාවිතය
එක් වරක් ලියා බොහෝ වර කැඳවන්න. එකම කේතය නැවත පිටපත් කිරීම අවශ්‍ය නැත.

කියවීමේ පහසුව
PrintLine, GetGrade වැනි නම් ප්‍රධාන වැඩසටහන තේරුම් ගැනීමට පහසු කරයි.

නඩත්තුවේ පහසුව
කාර්යයක් නිවැරදි කිරීමට හෝ වෙනස් කිරීමට එක් තැනක පමණක් වෙනස් කරන්න.

කණ්ඩායම් වැඩ සහ පරීක්ෂාව
විවිධ අයට විවිධ උප වැඩසටහන් ලියා පරීක්ෂා කළ හැක.

2. උප වැඩසටහන් වර්ග දෙක

	අගයක් ආපසු දෙන	අගයක් ආපසු නොදෙන
Pascal නම	function	procedure
Python	return අගයක් සහිත def	return අගයක් නැති def
ආපසු දෙන්නේ	එක් අගයක් (උදා ශ්‍රේණියක්)	කිසිවක් නැත – කාර්යයක් පමණක් කරයි
උදාහරණය	GetGrade(82) → 'A'	PrintLine

3. පරාමිති සහ තර්ක

උප වැඩසටහනකට යවන අගයන් **පරාමිති** (parameters) වේ. area(length, width) හි length සහ width පරාමිති වේ. area(5, 4) කැඳවන විට 5 සහ 4 අගයන් **තර්ක** (arguments) වේ.

4. අගයක් ආපසු දෙන ශ්‍රිතයක්

GetGrade ලකුණක් ලබාගෙන O/L ශ්‍රේණි සීමා (A 75, B 65, C 50, S 35) භාවිතයෙන් ශ්‍රේණි අකුරක් **ආපසු දෙයි**.

ව්‍යාජ කේතය (Pseudocode)

```
FUNCTION GetGrade(Marks)
  IF Marks >= 75 THEN RETURN "A"
  ELSE IF Marks >= 65 THEN RETURN "B"
  ELSE IF Marks >= 50 THEN RETURN "C"
  ELSE IF Marks >= 35 THEN RETURN "S"
  ELSE RETURN "W"
  ENDIF
ENDFUNCTION

BEGIN
  FOR k = 1 TO 3
    INPUT M
    DISPLAY GetGrade(M)
  ENDFOR
END
```

අගයක් ආපසු දෙන උප ක්‍රමලේඛය (function): GetGrade

Pascal

```
program GradeFunction;
var
  k, m: integer;

function GetGrade(marks: integer): char;
begin
  if marks >= 75 then
    GetGrade := 'A'
  else if marks >= 65 then
    GetGrade := 'B'
  else if marks >= 50 then
    GetGrade := 'C'
  else if marks >= 35 then
    GetGrade := 'S'
  else
    GetGrade := 'W';
end;

begin
  for k := 1 to 3 do
  begin
    write('Enter mark: ');
    readln(m);
    writeln('Grade: ', GetGrade(m));
  end;
end.
```

Python

```
def get_grade(marks):
    if marks >= 75:
        return "A"
    elif marks >= 65:
        return "B"
    elif marks >= 50:
        return "C"
    elif marks >= 35:
        return "S"
    else:
        return "W"

for k in range(3):
    m = int(input("Enter mark: "))
    print("Grade:", get_grade(m))
```

නියැදි ධාවනය

```
Enter mark: 82
Grade: A
Enter mark: 47
Grade: S
Enter mark: 66
Grade: B
```

> 💡 Pascal හි ප්‍රතිඵලය ලබා දෙන්නේ ශ්‍රිත නමට පැවරීමෙනි: GetGrade := 'A'. Python හි return "A" ලෙස ලියමු.

5. ක්‍රියා පටිපාටි (අගයක් ආපසු නොදෙන)

සාප්පු බිල්පතක PrintLine තුන් වරක් ද එක් එක් අයිතමය සඳහා ShowItem ද භාවිත කරයි. PrintLine ට **පරාමිති නැත**; ShowItem ට තුනක් ඇත.

අගයක් ආපසු නොදෙන උප ක්‍රමලේඛ (procedures): කඩ බිලක්

Pascal

```
program BillProcedures;

procedure PrintLine;
begin
  writeln('------------------------');
end;

procedure ShowItem(item: string; price, qty: integer);
begin
  writeln(item, ': Rs. ', price * qty);
end;

begin
  PrintLine;
  writeln('SAMPLE SHOP BILL');
  PrintLine;
  ShowItem('Bread', 120, 2);
  ShowItem('Milk packet', 280, 1);
  ShowItem('Eggs', 55, 6);
  PrintLine;
end.
```

Python

```
def print_line():
    print("-" * 24)

def show_item(item, price, qty):
    print(item + ": Rs.", price * qty)

print_line()
print("SAMPLE SHOP BILL")
print_line()
show_item("Bread", 120, 2)
show_item("Milk packet", 280, 1)
show_item("Eggs", 55, 6)
print_line()
```

නියැදි ධාවනය

```
------------------------
SAMPLE SHOP BILL
------------------------
Bread: Rs. 240
Milk packet: Rs. 280
Eggs: Rs. 330
------------------------
```

6. ශ්‍රිතයක් බොහෝ වර කැඳවීම

ශ්‍රිතයක් සංඛ්‍යාවක් මෙන්ම ප්‍රකාශනයක් තුළ භාවිත කළ හැක.

පරාමිති දෙකක් සහිත function: සෘජුකෝණාස්‍රයක වර්ගඵලය

Pascal

```
program RectArea;

function Area(length, width: integer): integer;
begin
  Area := length * width;
end;

begin
  writeln('Room 1 = ', Area(5, 4));
  writeln('Room 2 = ', Area(6, 3));
  writeln('Both = ', Area(5, 4) + Area(6, 3));
end.
```

Python

```
def area(length, width):
    return length * width

print("Room 1 =", area(5, 4))
print("Room 2 =", area(6, 3))
print("Both =", area(5, 4) + area(6, 3))
```

නියැදි ධාවනය

```
Room 1 = 20
Room 2 = 18
Both = 38
```

💡 ඔබ දැනටමත් භාවිත කරන ගොඩනැගූ උප වැඩසටහන්: writeln, readln, sqrt (Pascal); print, input, len, round (Python).

ප්‍රධාන පද

උප වැඩසටහන – එක් කාර්යයක් කරන, නමක් ඇති වැඩසටහන් කොටසක්.

ශ්‍රිතය – අගයක් ආපසු දෙන උප වැඩසටහනක්.

ක්‍රියා පටිපාටිය – කාර්යයක් කරන නමුත් අගයක් ආපසු නොදෙන උප වැඩසටහනක්.

කැඳවීම – නම ලිවීමෙන් උප වැඩසටහනක් භාවිත කිරීම.

පරාමිතිය – උප වැඩසටහනකට යවන අගයක් ලබාගන්නා විචල්‍යය.

තර්කය – කැඳවන විට යවන සත්‍ය අගය, උදා area(5, 4) හි 5.

ආපසු අගය – ශ්‍රිතයක් ආපසු දෙන ප්‍රතිඵලය.

නැවත භාවිතය – කේතය එක් වරක් ලියා බොහෝ වර භාවිතය.

කියවීමේ පහසුව – කේතය කියවීමට හා තේරුම් ගැනීමට ඇති පහසුව.

නඩත්තුවේ පහසුව – කේතය නිවැරදි කිරීමට හා වෙනස් කිරීමට ඇති පහසුව.

def – උප වැඩසටහනක් නිර්වචනය කිරීමට Python මූල පදය.

ගොඩනැගූ ශ්‍රිතය – භාෂාව විසින්ම සපයන ශ්‍රිතයක්, උදා len.

ආදර්ශ ප්‍රශ්න සහ පිළිතුරු

1. උප වැඩසටහනක් යනු කුමක්ද?

> **පිළිතුර:** එක් කාර්යයක් කරන, අවශ්‍ය විට කැඳවිය හැකි නමක් ඇති වැඩසටහන් කොටසකි.

2. උප වැඩසටහන්වල වාසි තුනක් දෙන්න.

> **පිළිතුර:** නැවත භාවිතය, කියවීමේ පහසුව සහ නඩත්තුවේ පහසුව.

3. ශ්‍රිතයක් සහ ක්‍රියා පටිපාටියක් අතර වෙනස?

> **පිළිතුර:** ශ්‍රිතයක් අගයක් ආපසු දෙයි; ක්‍රියා පටිපාටියක් එසේ නොකරයි.

4. GetGrade(66) කුමක්ද?

> **පිළිතුර:** B.

5. area(6, 3) ආපසු දෙන්නේ කුමක්ද?

> **පිළිතුර:** 18.

6. area(5, 4) හි 5 සහ 4 යනු මොනවාද?

> **පිළිතුර:** තර්ක (length සහ width පරාමිති සඳහා අගයන්).

බහුවරණ ප්‍රශ්න

1. එක් කාර්යයක් කරන නමක් ඇති වැඩසටහන් කොටසක්...

A. දත්ත වර්ගයකි
B. විවරණයකි
C. නියතයකි
D. උප වැඩසටහනකි

2. අගයක් ආපසු දෙන උප වැඩසටහනක්...

A. ශ්‍රිතයකි
B. අරාවකි
C. ක්‍රියා පටිපාටියකි
D. ලූපයකි

3. Pascal හි අගයක් ආපසු නොදෙන උප වැඩසටහනක්...

A. record
B. const
C. procedure
D. function

4. “එක් වරක් ලියන්න, බොහෝ වර භාවිත කරන්න” විස්තර කරන්නේ...

A. කැඳැලි කිරීම
B. සම්පාදනය
C. නැවත භාවිතය
D. දෝෂ නිරාකරණය

5. කාර්යයක් එක් තැනක පමණක් නිවැරදි කිරීම පෙන්වන්නේ...

A. නඩත්තුවේ පහසුව
B. පුනරාවර්තනය
C. සෘජු ප්‍රවේශය
D. ප්‍රමුඛතාව

6. PrintLine වැනි පැහැදිලි නම් වැඩි දියුණු කරන්නේ...

A. කියවීමේ පහසුව
B. මුද්‍රකයේ වේගය
C. දත්ත වර්ගය
D. මතක ප්‍රමාණය

7. GetGrade(82) ආපසු දෙන්නේ...

A. S
B. B
C. C
D. A

8. GetGrade(47) ආපසු දෙන්නේ...

A. C
B. S
C. W
D. B

9. area(5, 4) + area(6, 3) සමාන වන්නේ...

A. 18
B. 20
C. 9
D. 38

10. def area(length, width): හි length සහ width...

A. වෙන් කළ වචන
B. පරාමිති
C. අරා
D. නියත

11. උප වැඩසටහනක් නිර්වචනය කරන Python මූල පදය...

A. func
B. proc
C. def
D. sub

12. Python හි ශ්‍රිතයක් ප්‍රතිඵලය ආපසු දෙන්නේ...

A. break
B. return
C. input
D. print

13. Pascal හි ශ්‍රිතයක් අගයක් ආපසු දෙන්නේ...

A. ශ්‍රිත නමට පැවරීමෙන්
B. print භාවිතයෙන්
C. end. ලිවීමෙන්
D. readln භාවිතයෙන්

14. ShowItem('Eggs', 55, 6) පෙන්වන්නේ...

A. Eggs: Rs. 55
B. Eggs: Rs. 330
C. Eggs: Rs. 6
D. Eggs: Rs. 61

15. ගොඩනැගූ Python ශ්‍රිතයක් කුමක්ද?

A. PrintLine
B. ShowItem
C. len
D. GetGrade

ව්‍යුහගත ප්‍රශ්න

1. උප වැඩසටහන් භාවිතයේ වාසි තුනක් සඳහන් කරන්න. (ලකුණු 3) *(ලකුණු 3)*

> නැවත භාවිතය – එක් වරක් ලියා බොහෝ වර කැඳවීම; කියවීමේ පහසුව – තේරුම් ගැනීමට පහසු; නඩත්තුව – එක් තැනක වෙනස් කිරීම (එකට 1).

2. අගයක් ආපසු දෙන සහ ආපසු නොදෙන උප වැඩසටහන් එක් එකකට උදාහරණයක් සමඟ වෙන්කර දක්වන්න. (ලකුණු 4) *(ලකුණු 4)*

> අගයක් ආපසු දෙන (ශ්‍රිතය) අගයක් ලබා දෙයි, උදා GetGrade (2); ආපසු නොදෙන (ක්‍රියා පටිපාටිය) කාර්යයක් පමණක් කරයි, උදා PrintLine (2).

3. ලකුණක් ලබාගෙන O/L ශ්‍රේණිය A, B, C, S හෝ W ආපසු දෙන ශ්‍රිතයක් (Pascal හෝ Python) ලියන්න. (ලකුණු 5) *(ලකුණු 5)*

පරාමිතිය සහිත ශ්‍රිත ශීර්ෂය (1); IF ≥ 75 A (1); ELSE IF ≥ 65 B, ≥ 50 C (1); ≥ 35 S නැතහොත් W (1); ප්‍රතිඵලය return / පැවරීම (1).

අගයක් ආපසු දෙන උප ක්‍රමලේඛය (function): GetGrade

Pascal

```
program GradeFunction;
var
  k, m: integer;

function GetGrade(marks: integer): char;
begin
  if marks >= 75 then
    GetGrade := 'A'
  else if marks >= 65 then
    GetGrade := 'B'
  else if marks >= 50 then
    GetGrade := 'C'
  else if marks >= 35 then
    GetGrade := 'S'
  else
    GetGrade := 'W';
end;

begin
  for k := 1 to 3 do
  begin
    write('Enter mark: ');
    readln(m);
    writeln('Grade: ', GetGrade(m));
  end;
end.
```

Python

```
def get_grade(marks):
    if marks >= 75:
        return "A"
    elif marks >= 65:
        return "B"
    elif marks >= 50:
        return "C"
    elif marks >= 35:
        return "S"
    else:
        return "W"

for k in range(3):
    m = int(input("Enter mark: "))
    print("Grade:", get_grade(m))
```

නියැදි ධාවනය

```
Enter mark: 82
Grade: A
Enter mark: 47
Grade: S
Enter mark: 66
Grade: B
```

4. def area(length, width): return length * width
print(area(5, 4) + area(6, 3)) මගින් පෙන්වන්නේ කුමක්ද? පරාමිති සහ තර්ක නම් කරන්න. (ලකුණු 3) *(ලකුණු 3)*

> 38 (1); පරාමිති: length, width (1); තර්ක: 5, 4, 6, 3 (1).

5. අයිතම නමක්, මිලක් සහ ප්‍රමාණයක් ලබාගෙන අයිතම මුළු මුදල පෙන්වන ShowItem ක්‍රියා පටිපාටියක් ලියන්න. එක් කැඳවීමක් පෙන්වන්න. (ලකුණු 4) *(ලකුණු 4)*

පරාමිති තුනක් සහිත ශීර්ෂය (1); price × qty ගණනය (1); පෙන්වීම (1); කැඳවීම උදා ShowItem('Bread', 120, 2) (1).

අගයක් ආපසු නොදෙන උප ක්‍රමලේඛ (procedures): කඩ බිලක්

Pascal

```
program BillProcedures;

procedure PrintLine;
begin
  writeln('------------------------');
end;

procedure ShowItem(item: string; price, qty: integer);
begin
  writeln(item, ': Rs. ', price * qty);
end;

begin
  PrintLine;
  writeln('SAMPLE SHOP BILL');
  PrintLine;
  ShowItem('Bread', 120, 2);
  ShowItem('Milk packet', 280, 1);
  ShowItem('Eggs', 55, 6);
  PrintLine;
end.
```

Python

```
def print_line():
    print("-" * 24)

def show_item(item, price, qty):
    print(item + ": Rs.", price * qty)

print_line()
print("SAMPLE SHOP BILL")
print_line()
show_item("Bread", 120, 2)
show_item("Milk packet", 280, 1)
show_item("Eggs", 55, 6)
print_line()
```

නියැදි ධාවනය

```
------------------------
SAMPLE SHOP BILL
------------------------
Bread: Rs. 240
Milk packet: Rs. 280
Eggs: Rs. 330
------------------------
```

සාරාංශය

- උප වැඩසටහන = එක් කාර්යයක් කරන නමක් ඇති කොටස.
- වාසි: නැවත භාවිතය, කියවීමේ පහසුව, නඩත්තුව.
- ශ්‍රිතය අගයක් ආපසු දෙයි.
- ක්‍රියා පටිපාටිය අගයක් ආපසු නොදෙයි.
- පරාමිති ලබා ගනී; තර්ක යවයි.
- Python: def … return.
- Pascal: FunctionName := value.
- ගොඩනැගූ: print, len, writeln, sqrt.

MCQ පිළිතුරු – 1.10

1. **D** – උප වැඩසටහන.
2. **A** – ශ්‍රිතය.
3. **C** – Procedure.
4. **C** – නැවත භාවිතය.
5. **A** – නඩත්තුව.
6. **A** – කියවීමේ පහසුව.
7. **D** – 82 ≥ 75.
8. **B** – 35 ≤ 47 < 50.
9. **D** – 20 + 18.
10. **B** – පරාමිති.
11. **C** – def.
12. **B** – return.
13. **A** – GetGrade := 'A'.
14. **B** – 55 × 6.
15. **C** – ගොඩනැගූ.

© 2026 schoolict.net (Ruwan Mangala Suraweera). සියලු හිමිකම් ඇවිරිණි.