

Naga College Foundation, Inc.

M.T Villanueva Ave. Naga City
College of Computer Studies

COMPUTER PROGRAMMING - 1

IT102 / IS102

Name: _____ Course/Year/Sec. _____ Date: _____

Retrieval Practice / Knowledge Check Activity 1 - Debugging and Syntax Analysis

- Please use only a **black ballpoint pen** when providing your answers.
- **No erasing or making corrections** is permitted. Be sure to review your work thoroughly before submitting your final answer.
- It's important to uphold honesty and integrity. **Any form of cheating is strictly forbidden.**

Multiple Choice [50 items]

Instructions:

- ☒ Shade the appropriate circle on the answer sheet.
- ☒ Select only the most appropriate

1.

```
int num;  
  
scanf("%f", &num);
```

What type of error occurs?

- A. Logical error
- B. Syntax error
- C. Input mismatch
- D. Runtime crash

2.

```
int x=5;  
  
if(x=0)  
  
    printf("Zero");  
  
else  
  
    printf("Non-zero");
```

Output?

- A. Zero
- B. Non-zero
- C. Compile error
- D. None

3.

```
printf("%d", 5/2);
```

Output?

- A. 2.5 B. 2 C. 3 D. 2.0

4.

```
int a=2,b=3;

printf("%d", a+++b);
```

Result?

A. 6 B. Undefined behavior C. 5 D. Error

5.

```
if(a>0);

    printf("Positive");
```

What happens?

A. Always executes regardless of condition B. Never executes C. Syntax error D. Logical flaw

6.

Which of the following variable declarations is invalid?

A. `int 1num=5;` B. `int num_1=5;` C. `float rate=0.5;` D. `char c='A';`

7.

```
printf("%d", sizeof('A'));
```

Output on 32-bit system?

A. 1 B. 2 C. 4 D. Depends

8.

```
int n=3;
```

```
if(n>1)

    if(n>2)

        printf("A");

    else

        printf("B");
```

Output?

A. A B. B C. Error D. None

9.

Identify the logical flaw:

```
if(score>90)

    printf("A");

else if(score>80)

    printf("B");

else if(score>70)

    printf("C");

else if(score>90)

    printf("D");
```

A. Redundant condition B. Missing braces C. No default clause D. Missing semicolon

10.

```
int i=0;

while(i<3);
```

```
printf("%d",i);
```

What is wrong with this code?

- A. Infinite loop B. Syntax error C. Missing increment D. None

11.

Which will cause a **runtime error**?

- A. `int a=5,b=0; printf("%d",a/b);`
B. `int x=10,y=2;`
`printf("%d",x/y);`
C. `printf("Done");`
D. `return 0;`

12.

Which operator has **highest precedence**?

- A. * B. ++ C. && D. =

13.

```
int x=1;

printf("%d", ++x + x++);
```

Result?

- A. 4 B. 3 C. Undefined D. 2

14.

```
int x=10;

if(x>5)

    if(x>8)

        printf("Large");

    else

        printf("Small");
```

Output?

- A. Large B. Small C. None D. Error

15.

Which statement correctly reads an integer input?

- A. `scanf("%d",&num);`
B. `scanf("%f",&num);`
C. `scanf("%c",&num);`
D. `scanf("%s",&num);`

16.

```
for(int i=0;i<5;i++);

    printf("%d",i);
```

Why is this incorrect?

- A. Semicolon ends loop prematurely B. Missing braces C. Variable scope error D. Infinite loop

17.

What type of error occurs when a semicolon is missing?

- A. Syntax error
B. Logical error
C. Runtime error
D. Semantic error

18.

Which keyword terminates a `switch` case?

- A. `break` B. `exit` C. `stop` D. `continue`

19.

```
if(a>0)

    if(b>0)

        printf("X");

else

    printf("Y");
```

Output if a=1, b=-1?

A. X B. Y C. None D. Error

20.

```
int a;

if(a>10)

    printf("High");
```

What is wrong?

- A. Variable not initialized
- B. Missing semicolon
- C. Wrong operator
- D. Infinite loop

21.

```
int n=5;

for(int i=1;i<=n;i++)

    printf("%d ",i*i);
```

Output?

A. 1 4 9 16 25 B. 1 2 3 4 5 C. 25 16 9 4 1 D. 0 1 4 9 16

22.

```
int sum=0;

for(int i=1;i<=4;i++)

    sum+=i;

printf("%d",sum);
```

A. 9 B. 10 C. 11 D. 15

23.

```
int i=5;

while(i--)

    printf("%d",i);
```

Output?

A. 43210 B. 54321 C. 3210 D. 4321

24.

```
for(int i=1;i<=3;i++)

    for(int j=1;j<=i;j++)

        printf("*");
```

How many * printed?

A. 6 B. 5 C. 7 D. 9

25.

Find the logical error:

```
int i=10;

while(i>=0)

    printf("%d",i++);
```

A. Infinite loop B. Missing decrement C. Wrong operator D. Invalid condition

26.

```
int i=1;

do{

    printf("%d",i);

}while(i>1);
```

Output?

A. 1 B. 0 C. None D. Infinite loop

27.

```
int n=4,sum=0;

for(int i=1;i<=n;i++) sum+=i*i;

printf("%d",sum);
```

Output?

A. 30 B. 20 C. 25 D. 16

28.

If $x=2$, $y=3$, $z=4$, evaluate $x+y*z$.

A. 14 B. 20 C. 18 D. 24

29.

```
int x=10;
```

```
while(x>0) x=x/2;
```

```
printf("%d",x);
```

Output?

A. 0

B. 1

C. 2

D. Infinite loop

30.

```
if(a>5)

    if(b<2)

        printf("X");

    else

        printf("Y");
```

When $a=6$, $b=1$, output?

A. X B. Y C. None D. Error

31.

Which loop guarantees at least one execution?

A. do-while B. for C. while D. nested for

32.

Which operator tests equality?

A. == B. = C. != D. >=

33.

```
for(int i=1;i<=5;i++){
```

```

    if(i==3) continue;

    printf("%d",i);

}

```

Output?

A. 1245 B. 12345 C. 1234 D. 125

34.

```

int x=0;

for(int i=1;i<=3;i++) x+=i;

printf("%d",x);

```

A. 3 B. 5 C. 6 D. 10

35.

```

int a=3;

if(a%2==0) printf("Even"); else
printf("Odd");

```

Output?

A. Even B. Odd C. None D. Error

36.

Which expression increments variable before use?

A. ++x B. x++ C. --x D. x--

37.

What will cause an infinite loop?

A. Missing update statement in `while` loop
B. Missing semicolon

C. Using wrong variable
D. `break` statement inside loop

38.

```

if(score>=75)

    printf("Passed");

else

    printf("Failed");

```

If `score=80`, output?

A. Passed B. Failed C. None D. Error

39.

Which logical operator ensures both conditions must be true?

A. && B. || C. ! D. =

40.

What happens when `break` executes inside nested loops?

A. Exits inner loop only
B. Exits all loops
C. Skips next iteration
D. Restarts loop

41.

To prevent division by zero:

A. Add `if(den!=0)` before division B. Use `continue` C. Add `break` D. Change denominator

42.

Find the error:

```
for(int i=0;i<10;i--)  
  
    printf("%d",i);
```

- A. Infinite loop
- B. Syntax error
- C. Missing braces
- D. Missing semicolon

43.

```
int a=5,b=10;  
  
a^=b; b^=a; a^=b;
```

Purpose?

- A. Swap values without temporary variable
- B. Reset values to 0
- C. Multiply a and b
- D. Compare a and b

44.

Why are arrays zero-indexed?

- A. Efficient address calculation in memory
- B. Easier to read
- C. Historical convention
- D. Compiler rule only

45.

Best method to locate logic bug?

- A. Dry-run or trace code manually

B. Guess error

C. Add random print statements

D. Restart compiler

46.

Effective debugging tool in IDEs:

- A. Step-by-step execution and breakpoints
- B. Syntax highlighting
- C. Auto-completion
- D. Compiler optimization

47.

Using a variable before initialization causes:

- A. Undefined or garbage value
- B. Compiler crash
- C. Runtime error always
- D. Correct output

48.

Effect of missing semicolon after `for` header?

- A. Skips loop body
- B. Infinite loop
- C. Compilation error
- D. Exits immediately

49.

What does `%` operator do?

- A. Returns remainder of division
- B. Multiplies two numbers

C. Performs modulo addition

D. Returns quotient

50.

Main advantage of using functions in code:

A. Improves reusability and readability

B. Slows execution

C. Avoids variables

D. Forces recursion