

UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 1, Examination, July 2017

Module No: EE1102      Module Name: Introduction to Programming

## Part I

---

Instructions for candidates

- Write your index number on top of every page.
- Question paper contains 50 multiple choice questions.
- Each question carries 0.4 marks.
- Answer all questions. Each question has only one answer.
- Read the question and all answers before making the choice.
- For each question, put an X mark on the letter: (a), (b), (c), or (d) which corresponds to the correct answer, by using a black or blue pen.
- Time allowed is 1 hour and 30 minutes.

- 
- |  |  |
|--|--|
| 1. Founder of the C language is  | 4. The command <code>gcc xyz.c</code> will produce the executable file |
| (a) Dennis Ritchie   | (a) <code>out.exe</code>   |
| (b) Bill Gates   | (b) <code>out.a</code>   |
| (c) Roger Moor   | (c) <code>a.out</code>   |
| (d) Steve Jobs   | (d) <code>xyz.o</code>   |
| 2. The <code>gcc</code>  | 5. A program with a syntax error                                       |
| (a) is used to compile console based applications  | (a) can be executed  |
| (b) can link library code to the source code   | (b) cannot be compiled   |
| (c) can not give run time error messages   | (c) can be executed, but an error message will be displayed            |
| (d) is correctly described by all above answers  | (d) will produce wrong results during execution                        |
| 3. The command <code>gcc -lm myprog.c</code> at the Linux prompt implies that                        | 6. The code  |
| (a) <code>myprog.c</code> may be the source file and it uses <code>stdio.h</code> library functions. | <code>x = y * 3</code>   |
| (b) <code>myprog.c</code> may contain mathematical functions   | <code>z = x + y * X * 3;</code>  |
| (c) <code>myprog.x</code> is the executable file   | contains   |
| (d) <code>myprog.o</code> is the object file   | (a) a syntax error   |
|  | (b) a run-time error   |
|  | (c) no errors  |
|  | (d) error or not depends on y value                                    |

7. The program
- ```
#include <stdio.h>
int main(void){/* printf("XYZ"); */
return 0;}
```
- (a) prints XYZ  
(b) prints nothing  
(c) has syntax errors  
(d) creates run time *memory segment* fault
8. Type of a variable does not define
- (a) the size of the memory required to hold data.  
(b) possible operations on variables of considered type.  
(c) possible pointers and references for data.  
(d) the kind of data to be stored
9. The preprocessor directive
- ```
#define VAR 3
defines a
```
- (a) variable  
(b) symbolic constant  
(c) library file  
(d) class
10. Which of the following declarations of x best fits to store a student's name?
- (a) char x;  
(b) char x[100];  
(c) int \*x;  
(d) float x[499].
11. C variable type that does not define any particular type is
- (a) float  
(b) int  
(c) void  
(d) char
12. What is given in the following?
- ```
int fun[10][11];
```
- (a) Declaration of two dimensional array fun  
(b) Definition of the operator [ ][ ].  
(c) Type casting of int.  
(d) Definition of function fun
13. In the statement
- ```
myConst = 3 * myVar;
```
- (a) myConst must be a variable.  
(b) myConst must be a literal constant  
(c) myVar must be a variable  
(d) myVar must be a function
14. Array of characters ending in null character is called
- (a) a structure.  
(b) an empty array  
(c) an array of ASCII characters.  
(d) a C-string.
15. What is the meaning of  $x = y$ ; ?
- (a) Value of y is assigned to the x  
(b) x is equal to y.  
(c) Is the x equal to y?  
(d) Is x is not equal to y?
16. What is the meaning of  $x == y$ ; ?
- (a) Value of y is assigned to the x.  
(b) x is equal to y  
(c) Is the x equal to y?  
(d) Is x is not equal to y?
17. what is the meaning of  $x != y$ ; ?
- (a) Value of y is assigned to the x.  
(b) x is equal to y  
(c) Is the x equal to y?  
(d) Is x is not equal to y?
18. What does the operation  $52\%3$  produce?
- (a) 0  
(b) 1  
(c) 2  
(d) 3
19. What does the operation  $150/15$  produce?
- (a) 0.01  
(b) 1.5

- (c) 10  
(d) 165
20. The expression `18 != 8` evaluates to
- (a) 1  
(b) 0  
(c) 10  
(d) 26
21. The expression `132 == 123` evaluates to
- (a) 1  
(b) 0  
(c) 6  
(d) 10
22. After execution of  
`x=3; ++x; ++x;`  
the value of `x` is
- (a) 2  
(b) 3  
(c) 4  
(d) 5
23. After execution of  
`x=3; x += x;`  
the value of `x` is
- (a) 4  
(b) 5  
(c) 6  
(d) 7
24. The expression `(x + y * 4) > 25` evaluates to 1 if
- (a) `x = 8` and `y = 4`  
(b) `x = 2` and `y = 6`  
(c) `x = 11` and `y = 3`  
(d) `x = 3` and `y = 5`
25. The expression `(5 == 55 || 3 != 3)` evaluates to
- (a) 1  
(b) 0  
(c) -1
- (d) 53
26. The expression `(10 += 4 && 5 < 3)` evaluates to
- (a) 1  
(b) 0  
(c) -1  
(d) Non of the above choices
27. `if(x) printf("Yes");`  
This displays Yes, if only if `x` is
- (a) greater than 0 or less than 0.  
(b) greater than 0.  
(c) less than 0.  
(d) 0.
28. `if(!x) printf("Yes");`  
This displays Yes only if `x` is
- (a) greater than 0 or less than 0.  
(b) greater than 0.  
(c) is less than 0.  
(d) 0.
29. `if( x+10 > 10) printf("Yes");`  
This displays Yes if `x` is
- (a) less than 10.  
(b) greater than 10.  
(c) greater than or equal to 0.  
(d) greater than 0.
30. `if( x-10 < 10) printf("Yes");`  
This displays Yes if `x` is
- (a) greater than 30.  
(b) greater than 10.  
(c) less than 40.  
(d) less than 20.
31. `if( x > 10 && x <=25) printf("Yes");`  
This displays Yes if `x` is
- (a) 0  
(b) 10  
(c) 25  
(d) 35
32. `if(x<=150 || x>175) printf("Yes");`  
This displays Yes if `x` is
- (a) 150

- (b) 160  
(c) 170  
(d) 175
33. `if((x<10 || x>25) && (x<-10 || x>-25))  
printf("Yes");`  
This displays Yes if x is  
(a) -15  
(b) 0  
(c) -30  
(d) any of the above.
34. `(12==5 && 3!=3) || (4+5 || 3-4+1)`  
This expression evaluates to  
(a) -1  
(b) 0  
(c) 1  
(d) non of the above values.
35. `for(i=0;i<10;++i) printf("X");`  
How many times the character X is displayed?  
(a) 9  
(b) 10  
(c) 11  
(d) 0
36. `for(i=0;i<10; i += 2) printf("X");`  
How many times the character X is displayed?  
(a) 2  
(b) 3  
(c) 4  
(d) 5
37. `for(i=25; i<10; i -= 2) printf("X");`  
How many times the character X is displayed?  
(a) 0  
(b) 12  
(c) 15  
(d) 5
38. `i=0;while(i<5){printf("%d",i);++i;}`  
This code prints  
(a) 0 1 2 3 4  
(b) 0 1 2 3 4 5  
(c) 1 2 3 4  
(d) 1 2 3 4 5
39. `i=0;while(i<5)++i;printf("%d",i);`  
The value of i displayed is  
(a) 3  
(b) 5  
(c) 7  
(d) undefined
40. `Sum=0;i=1;while(i<=5){Sum+=i;++i;}`  
This code sets Sum to  
(a) 13  
(b) 14  
(c) 15  
(d) 16
41. `In do{Statement;} while(Condition);`  
(a) Condition is evaluated only if Statement is not null.  
(b) Statement is executed before Condition  
(c) Condition is evaluated before Statement.  
(d) Statement is executed only if Condition is true.
42. In switch statement  
(a) goto can be used to direct to another case.  
(b) default is not optional.  
(c) break prevents execution of next case.  
(d) continue makes execution of previous case.
43. `a=1;b=1; f=1;  
while(a<=14)  
{ f = a + b;  
  b=a; a=f;  
  printf(" %d ", f);  
}`  
Above code displays  
(a) 2 3 6 9 14 24  
(b) 2 3 3 11 15 23  
(c) 2 3 5 8 13 21

- (d) 2 3 6 10 22 71
44. 

```
for(i=0, j=10; i<35; i+=j, --j)
{ printf(" %d ", j); }
```

 Above code displays
- (a) 10 9 8 7 6  
 (b) 10 9 7 6 5  
 (c) 10 8 8 6 5  
 (d) 10 9 5 3 2
45. 

```
i=1000;
while(i)
{ i-=2;
  printf("\n%d", i);
}
```

 Above code displays
- (a) odd numbers.  
 (b) even numbers.  
 (c) fractional numbers.  
 (d) prime numbers.
46. 

```
i=10;
while(i-->0)
{
  j=10; while(j-->0) printf("X");
  printf("\n");
}
```

 The "shape", made out of X, displayed by the above code is a
- (a) square.  
 (b) circle.  
 (c) triangle.  
 (d) non of the above.
47. 

```
void pow(void);
```

 is a function
- (a) prototype  
 (b) definition  
 (c) call  
 (d) name
48. According to 

```
float xy(int x, float y);
```

 the return value is
- (a) x  
 (b) y  
 (c) combination of x and y.  
 (d) of type float.
49. Local variables of a function
- (a) can be define during run-time.  
 (b) do exist only during the function execution.  
 (c) sometimes can act as global variables.  
 (d) are constants and can not be changed.
50. If  $y=7$ , after calling the function as  $z=\text{add3}(y)$ , where
- ```
int add3(int x)
{ return x+x+x;
}
```
- value of
- $z$
- is
- (a) 21  
 (b) 7  
 (c) 22  
 (d) non of the above.