UNIVERSITY OF RUHUNA

Faculty of Engineering

End-Semester 1, Examination, October 2019

Module No: EE1101 Module Name: Computer Programming I

Part-A

[1 hour and 30 minutes]

Instructions for candidates

- Write your index number on top of every page.
- Question paper contains 50 multiple choice questions.
- Answer all questions. Each question has only one answer.
- **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.**
- Each correct answer carries 0.4 marks.

---

1. The command `gcc`
   (a) is used to edit a c file.
   (b) is used to compile a c file.
   (c) is used to copy a file.
   (d) is used to create a c++ file.

2. The command `gcc -o myprog myprog.c` at the Linux prompt implies that
   (a) `myprog.c` may be the source file and it uses `stdio.h` library functions.
   (b) `myprog.c` may contain mathematical functions.
   (c) `myprog` is the result of the above command.
   (d) `myprog` will be used to compile `myprog.c`.

3. The command `gcc xyz.c` will produce the executable file
   (a) `out.exe`.
   (b) `xyz.out`.
   (c) `xyz.obj`.
   (d) `a.out`.

4. A program with a syntax error
   (a) can be executed.
   (b) cannot be compiled.
   (c) can be executed, but an error message will be displayed.
   (d) will produce wrong results during execution.

5. Which of the symbols is used to terminate c statement?
   (a) ;
   (b) '
   (c) "
   (d) \n

6. Which of the given words is not a keyword in C?
   (a) int
   (b) double
   (c) float
   (d) integer

7. When you compile the code
   `int x, y;`

`printf("%d", x = 3 );`
gives

(a) a syntax error .
(b) a run-time error.
(c) no errors.
(d) error or not depends on y value.

8. 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.

9. Which of the following is not a valid variable name declaration?

(a) int _a3;
(b) int a_3;
(c) int 3_a;
(d) int _3a

10. Which of the following is not a valid C variable name?

(a) int number;
(b) float rate;
(c) int variable_count;
(d) int $main;

11. Which of the given statements makes the pointer `int *p;` points to the variable `int x;`?

(a) p = &x;
(b) *p = x;
(c) *p !=x;
(d) p[x]=*x;

12. If the pointer `int *p;` points to x, then x can be assigned 10 by

(a) p =10;.
(b) p=&x;.
(c) *p = 10;.
(d) p=3; x = &p;.

13. Which of the format specifiers is used to print the values of integer type variable?

(a) %lf
(b) %d
(c) %s
(d) %c

14. Which of the given operators is an unary operator?

(a) -
(b) ++
(c) *
(d) +=

15. The preprocessor directive
`#include <stdlib.h>`
defines a

(a) variable.
(b) symbolic constant.
(c) library file.
(d) class.

16. Type of a variable defines

(a) the size of the memory required to hold data.
(b) possible operations on variables of considered type.
(c) the kind of data to be stored.
(d) all given by above answers.

17. Which of the following declarations of x best fits to store a name of a person?

(a) `char x;`
(b) `char x[100];`.
(c) `int *x;`.
(d) `float x[499];`.

18. C variable type that does not define any particular type is

(a) `float`
(b) `int`

(c) **void**

(d) **char**

19. What is given in the following?
`int fun[22];`

(a) Declaration of array fun with 22 elements

(b) Definition of the operator [ ].

(c) Declaration of array fun with 21 elements

(d) Definition of 22 functions under the name fun

20. What is the meaning of `x = y;` ?

(a) Assign the value of y to the x.

(b) x is equal to y.

(c) Is x equal to y?

(d) Is x is not equal to y?

21. 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?

22. What does the operation 23%4 produce?

(a) 3

(b) 2

(c) 1

(d) 0

23. What does the operation 150/15 produce?

(a) 0.01

(b) 1.5

(c) 10

(d) 165

24. The expression `25 != 24` evaluates to

(a) 1.

(b) 0.

(c) 10.

(d) 26.

25. The expression `1232 == 123` evaluates to

(a) 1.

(b) 0.

(c) 6.

(d) 8.

26. After execution of
`x=2; ++x; ++x;`
the value of x is

(a) 2.

(b) 3.

(c) 4.

(d) 5.

27. After execution of
`x=5; x += x;`
the value of x is

(a) 55.

(b) 5.

(c) 10.

(d) 25.

28. The expression `(x + y * 5) > 25` evaluates to 1, if

(a) x = 3 and y = 4.

(b) x = 0 and y = 5.

(c) x = 11 and y = 3.

(d) x = 14 and y = 2.

29. The expression `(3 == 33 || 3 != 3 )` evaluates to

(a) 33.

(b) 42.

(c) 1.

(d) 0.

30. The expression `(10 -= 4 && 5<3 )` evaluates to

(a) 1.

(b) 0.

(c) -1.

(d) none of the above choices.

31. `if(!x) printf("Yes");`
This displays Yes only if x is

(a) 0.

(b) greater than 0 or less than 0.

(c) is less than 0.

(d) greater than 0.

32. `if( x+10 > 10) printf("Yes");`
This displays Yes if x is

(a) greater than 10.

(b) less than 10.

(c) greater than or equal to 0.

(d) greater than 0.

33. `if( x-25 < 25) printf("Yes");`
This displays Yes if x is

(a) greater than 50.

(b) greater than 10.

(c) less than 40.

(d) less than 55.

34. Consider the following code.

```
if( x > 100 && x <=205)
 printf("Yes");
else
 printf("No");
```

This displays **No** if x is

(a) 100.

(b) 110.

(c) 125.

(d) 205.

35. Consider the following code.

```
if(x<=150 || x>175)
  printf("Yes");
else
  printf("No");
```

This displays **No** if x is

(a) 150.

(b) 160.

(c) 180.

(d) 148.

36. Consider the following code.

```
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.

37. `(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.

38. `for(i=0;i<610;++i) printf("X");`
How many times the character X is displayed?

(a) 69.

(b) 610.

(c) 161.

(d) 0.

39. `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

40. `for(i=20; i<10; i -= 2) printf("X");`
How many times the character X is displayed?

(a) 0

(b) 5

(c) 4

(d) 8

41. `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.

42. `i=0;while(i<4)++i;printf("%d",i);`
The value of i displayed is

(a) 0 1 2 3.

(b) 4.

(c) 3 2 1 0.

(d) Non of the above.

43. `Sum=4;i=2;while(i<=5){Sum+=i;++i;}`
This code sets Sum to

(a) 17.

(b) 16.

(c) 15.

(d) 18.

44. 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**.

45.
```
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.

46.
```
for(i=-5, j=11;i<34; i+=j,--j)
{ printf("%d ", j); }
```
Above code displays

(a) 11 10 9 8 7.

(b) 11 9 7 5 3.

(c) 10 8 8 6 -5.

(d) 10 9 5 -3 2.

47.
```
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.

48. `void tow(float *x);`
is a function

(a) definition.

(b) call.

(c) name.

(d) prototype.

49. According to
`double xy(int x, float y);`
the return value is

(a) of type **float**.

(b) of type **int**.

(c) of type doublet.

(d) of non of the types given above.

50. If `y=8`, after calling the function as `z=add3(y)`, where
```
int add3(int x)
{ return x+x+x;
}
```
the value of z is

(a) 8.

(b) 24.

(c) 888.

(d) non of the above.