

Index Number:



Sri Lanka Institute of Information Technology
B.Sc. Eng. (Honours) Degree

Final Examination
Year 1, Semester II (2016)

EC1441 – Engineering Programming

Duration: 2 Hours

October 2016

Instructions to Candidates:

- This paper has 2 **Sections**. Answer **ALL** questions, in both Sections.
- There are **20 questions** in the first Section.
 - Each question in this section carries 1 mark for a Total of 20 Marks.
- In Section 2, you have to **write 3 small programs**.
 - The first program is for 4 marks, the final two programs are worth 8 Marks each for a Total of 20 Marks.
- **This paper carries a total of 40 marks.**
- This paper contains 14 pages including the Cover Page, and back Page.
- Write your answers on this paper itself. If you need additional paper, please write you EN# on that paper and attach at the end of this paper.

page	2	3	4	5	6	7	8	9	10	total
marks										
11	12	13	14	15	16	17	18	19	20	

Version **A**

Section 1

(2 marks for each question)

Indicate your response(s) by clearly making a mark in front of your selection(s).

1) The following code:

```
#include <stdio.h>

void main(void){
    char a[]={11,22,33,44,55};
    char *ptr1;
    ptr1=a;
    printf("%d\n",*(ptr1+1));
}
```

Prints:

- A) 11 B) 12 C) 22 D) 33 E) Compile error

2) The following code:

```
#include <stdio.h>

void main(void){
    char a[]={11,22,33,44,55};
    char *ptr1;
    ptr1=a;
    printf("%d\n", (*ptr1+1));
}
```

Prints:

- A) 11 B) 12 C) 22 D) 33 E) Compile error

3) The following code:

```
#include <stdio.h>

void main(void){
    char a[]={11,22,33,44,55};
    printf("%c\n",*(++a));
}
```

Prints:

- A) 11 B) 12 C) 22 D) 33 E) Compile error

4) The following code:

```
#include <stdio.h>

void main(void){
    char a[]={11,22,33,44,55};
    char *ptr1;
    ptr1=a;
    printf("%d\n",*(++ptr1));
}
```

Prints:

- A) 11 B) 12 C) 22 D) 33 E) Compile error

5) Inspect the following code:

```
#include <stdio.h>

void main(void){
    FILE *fp;
    FILE *fp2;
    char ch;

    fp=stdin;
    fp2=fopen("log.txt", "w");

    while((ch=getchar())!=EOF){
        putchar(ch);
        fputc(ch,fp2);
    }
}
```

Select the option below relating to this code that is **NOT CORRECT**.

This code:

- A) prints every input character to the screen
- B) prints every input character to a file called "log.txt"
- C) Exit's the application when 'Control-D' is pressed at the key board.
- D) Crashes when run.

6) The following code:

```
#include <stdio.h>
void fl(int *i);

void main(void){
    int i=5;
    fl(&i);
    printf("i=%d\n",i);
}

void fl(int *i){
    (*i)+=1;
}
```

prints i=:

- A) 5 B) 6 C) 56 D) 65 E) Compile Error

7) The following code:

```
#include <stdio.h>
void fl(int *i);

void main(void){
    int i=5;
    int *ptr;
    ptr=&i;
    fl(ptr);
    printf("i=%d\n",i);
}

void fl(int *ptr){
    (*ptr)+=1;
}
```

prints i=:

- A) 5 B) 6 C) 56 D) 65 E) Compile Error

8) The following code:

```
#include <stdio.h>
void f1(int *i);

void main(void){
    int i=5;
    f1(&i);
    printf("i=%d\n",i);
}

void f1(int *i){
    *(i=i+1);           //look carefully
}
```

prints i=:

- A) 5 B) 6 C) 56 D) 65 E) Compile Error

9) The following code:

```
#include <stdio.h>

void main(void){
    char a[]="123456789";
    char b[]="Hello EC144";
    char *ptr1;
    ptr1=&(b[6]);        //look carefully
    ptr1==a;              //look carefully
    printf("%s\n",ptr1);
}
```

Prints:

- A) EC144
B) 123456789
C) Hello EC144
D) Hello
E) Compile error

10) The following code:

```
#include <stdio.h>
void main(void){
    char a[]="123456789";
    char b[]="Hello EC144";
    char *ptr1;
    ptr1=&(b[6]);          //look carefully
    ptr1=a;                //look carefully
    printf("%s\n",ptr1);
}
```

Prints:

- A) EC144
- B) 123456789
- C) Hello EC144
- D) Hello
- E) Compile error

11) The following code:

```
#include <stdio.h>

void main(void){
    char a[]="Hello EC144";
    char *ptr1;
    ptr1=a;
    printf("%c\n",*ptr1);
}
```

Prints:

- A) H
- B) I
- C) e
- D) l
- E) Compile error

12) The following code:

```
#include <stdio.h>

void main(void){
    char a[]="Hello EC144";
    char *ptr1;
    ptr1=a;
    printf("%c\n",*(ptr1+1));
}
```

Prints:

- A) H
- B) I
- C) e
- D) l
- E) Compile error

13) The following code:

```
#include <stdio.h>

void main(void){
    char a[]="Hello EC144";
    char *ptr1;
    ptr1=a;
    printf("%c\n",(*ptr1)+1);
}
```

Prints:

- A) H B) I C) e D) l E) Compile error

14) The following code:

```
#include <stdio.h>
struct point{
    int x;
    int y;
};
void main(void){
    struct point p1={1,2};
    struct point p2={3,4},p3={5,6};
    struct point *ptr;
    ptr=&p3;
    p3=p1;
    printf("%d,%d\n",ptr->x, ptr->y);
}
```

This code :

- A) prints : 1,2
B) prints : 3,4
C) prints : 5,6
D) causes compile errors.

15) Inspect the following code:

```
#include <stdio.h>
struct point{
    int x;
    int y;
};
void main(void){
    struct point p1={1,2};
    struct point p2={3,4},p3={5,6};
    struct point *ptr;
    p1=p3;
    ptr=p2;
    printf("%d,%d\n",(*ptr).x, (*ptr).y);
}
```

This code :

- A) prints : 1,2
- B) prints : 3,4
- C) prints : 5,6
- D) causes compile errors.

16) Inspect the following code:

```
#include <stdio.h>
#include <stdlib.h>
struct point{
    int x;
    int y;
};
void main(void){
    struct point p1={1,2};
    struct point p2={3,4},p3={5,6};
    struct point *ptr;
    ptr=(struct point *)malloc(sizeof(struct point));
    (*ptr)=p2;
    printf("%d,%d\n",(*ptr).x, (*ptr).y);
}
```

This code :

- A) prints : 1,2
- B) prints : 3,4
- C) prints : 5,6
- D) causes compile errors.

17) Inspect the following code:

```
#include <stdio.h>
void main(void){
    {
        int i=4;
        printf("i=%d, ",i);
    }
    printf("i=%d\n",i);
    int i=7;
}
```

This code:

- A) prints i=4,
- B) prints i=4, i=7
- C) prints i=4, i=4
- D) causes compile errors.

18) Inspect the following code:

```
#include <stdio.h>
int i=7;
void main(void){
    {
        int i=4;
        printf("i=%d, ",i);
    }
    printf("i=%d\n",i);
}
```

This code:

- A) prints i=4,
- B) prints i=4, i=7
- C) prints i=4, i=4
- D) causes compile errors.

19) Inspect the following code:

```
#include <stdio.h>
void main(void){
    int i=4;
    printf("i=%d, ",i);
    int i=7;
    printf("i=%d\n",i);
}
```

This code:

- A) prints i=4,
- B) prints i=4, i=7
- C) prints i=4, i=4
- D) causes compile errors.

20) Inspect the following code:

```
#include <stdio.h>
int i=7;
void main(void){
    {
        i=i+1;
        int i=4;
        printf("i=%d, ",i);
    }
    printf("i=%d\n",i);
}
```

This code:

- A) prints i=4, i=7
- B) prints i=5, i=7
- C) prints i=4, i=8
- D) causes compile errors.

- end of section-

Section 2 **(20 marks total, each program receives different marks.)**

(if you need additional space, please write the code on a separate paper, write your EN# and attach that paper to this paper)

- 1) Write a small program to print all odd numbers from 1 to 100. You should show all the code needed for this program, including any libraries. **(4 Marks)**

2) Write a program to sort N numbers stored in an array, in ascending order . Assume the following code is already provided. Start writing your code in main() below. **(8 Marks)**

```
#include <stdio.h>
```

```
int sort[]={23, 11, 56 , 2, 7, 21, 3, 1};
```

```
int count = 8;    // The number of values in the array to sort.
```

```
int main(void){
```

3) Write a program to copy values in one type of structure to another. Use the definitions listed below. (use the provided code as the start of your program) **(8 Marks)**

```
#include <stdio.h>
```

```
struct cord_3d{  
    int x;  
    int y;  
    int z;  
};
```

```
struct cord_2d{  
    int x;  
    int y;  
};
```

```
int main(void){  
    // assume cord_3d has been initialized as below:  
    struct cord_3d p3_1={1,1,1};  
    struct cord_3d p3_2={2,2,2};  
    struct cord_3d p3_3={3,3,3};  
  
    // now you need to write the code to allocate 3 variables for 'struct cord_2d'  
    // and copy the x and y values in p3_1, p3_2, p3_3 to your 3 new variables.  
    // finally write the code to print the values in each of the 3 variables you allocated.  
    // your out put should look like (1,1) followed by (2,2) followed by (3,3)
```

- end of Examination -

Index Number:



Sri Lanka Institute of Information Technology
B.Sc. Eng. (Honours) Degree

Final Examination
Year 1, Semester II (2016)

EC1441 – Engineering Programming

Duration: 2 Hours

October 2016

Instructions to Candidates:

- This paper has 2 **Sections**. Answer **ALL** questions, in both Sections.
- There are **20 questions** in the first Section.
 - Each question in this section carries 1 mark for a Total of 20 Marks.
- In Section 2, you have to **write 3 small programs**.
 - The first program is for 4 marks, the final two programs are worth 8 Marks each for a Total of 20 Marks.
- **This paper carries a total of 40 marks.**
- This paper contains 14 pages including the Cover Page, and back Page.
- Write your answers on this paper itself. If you need additional paper, please write you EN# on that paper and attach at the end of this paper.

page	2	3	4	5	6	7	8	9	10	total
marks										
11	12	13	14	15	16	17	18	19	20	

Version **B**

Section 1

(2 marks for each question)

Indicate your response(s) by clearly making a mark in front of your selection(s).

1) The following code:

```
#include <stdio.h>

void main(void){
    char a[]={11,22,33,44,55};
    char *ptr1;
    ptr1=a;
    printf("%d\n",*(++ptr1));
}
```

Prints:

- A) 11 B) 12 C) 22 D) 33 E) Compile error

2) The following code:

```
#include <stdio.h>

void main(void){
    char a[]={11,22,33,44,55};
    char *ptr1;
    ptr1=a;
    printf("%d\n",*(ptr1+1));
}
```

Prints:

- A) 11 B) 12 C) 22 D) 33 E) Compile error

3) The following code:

```
#include <stdio.h>

void main(void){
    char a[]={11,22,33,44,55};
    char *ptr1;
    ptr1=a;
    printf("%d\n",(*ptr1+1));
}
```

Prints:

- A) 11 B) 12 C) 22 D) 33 E) Compile error

4) The following code:

```
#include <stdio.h>

void main(void){
    char a[]={11,22,33,44,55};
    printf("%c\n",*(++a));
}
```

Prints:

- A) 11 B) 12 C) 22 D) 33 E) Compile error

5) Inspect the following code:

```
#include <stdio.h>

void main(void){
    FILE *fp;
    FILE *fp2;
    char ch;

    fp=stdin;
    fp2=fopen("log.txt", "w");

    while((ch=getchar())!=EOF){
        putchar(ch);
        fputc(ch,fp2);
    }
}
```

Select the option below relating to this code that is **NOT CORRECT**.

This code:

- A) prints every input character to the screen
- B) prints every input character to a file called "log.txt"
- C) Exit's the application when 'Control-D' is pressed at the key board.
- D) Crashes when run.

6) The following code:

```
#include <stdio.h>
void f1(int *i);

void main(void){
    int i=5;
    int *ptr;
    ptr=&i;
    f1(ptr);
    printf("i=%d\n",i);
}

void f1(int *ptr){
    (*ptr)+=1;
}
```

prints i=:

A) 5

B) 6

C) 56

D) 65

E) Compile Error

7) The following code:

```
#include <stdio.h>
void f1(int *i);

void main(void){
    int i=5;
    f1(&i);
    printf("i=%d\n",i);
}

void f1(int *i){
    *(i=i+1);           //look carefully
}
```

prints i=:

A) 5

B) 6

C) 56

D) 65

E) Compile Error

8) The following code:

```
#include <stdio.h>
void fl(int *i);

void main(void){
    int i=5;
    fl(&i);
    printf("i=%d\n",i);
}

void fl(int *i){
    (*i)+=1;
}
```

prints i=:

- A) 5 B) 6 C) 56 D) 65 E) Compile Error

9) The following code:

```
#include <stdio.h>

void main(void){
    char a[]="123456789";
    char b[]="Hello EC144";
    char *ptr1;
    ptr1=&(b[6]);           //look carefully
    ptr1==a;                //look carefully
    printf("%s\n",ptr1);
}
```

Prints:

- A) EC144
B) 123456789
C) Hello EC144
D) Hello
E) Compile error

10) The following code:

```
#include <stdio.h>

void main(void){
    char a[]="Hello EC144";
    char *ptr1;
    ptr1=a;
    printf("%c\n",*(ptr1+1));
}
```

Prints:

- A) H B) I C) e D) l E) Compile error

11) The following code:

```
#include <stdio.h>

void main(void){
    char a[]="Hello EC144";
    char *ptr1;
    ptr1=a;
    printf("%c\n",(*ptr1)+1);
}
```

Prints:

- A) H B) I C) e D) l E) Compile error

12) The following code:

```
#include <stdio.h>
void main(void){
    char a[]="123456789";
    char b[]="Hello EC144";
    char *ptr1;
    ptr1=&(b[6]);           //look carefully
    ptr1=a;                 //look carefully
    printf("%s\n",ptr1);
}
```

Prints:

- A) EC144
B) 123456789
C) Hello EC144
D) Hello
E) Compile error

13) The following code:

```
#include <stdio.h>

void main(void){
    char a[]="Hello EC144";
    char *ptr1;
    ptr1=a;
    printf("%c\n",*ptr1);
}
```

Prints:

A) H

B) I

C) e

D) l

E) Compile error

14) The following code:

```
#include <stdio.h>
struct point{
    int x;
    int y;
};
void main(void){
    struct point p1={1,2};
    struct point p2={3,4},p3={5,6};
    struct point *ptr;
    ptr=&p3;
    p3=p1;
    printf("%d,%d\n",ptr->x, ptr->y);
}
```

This code :

A) prints : 1,2

B) prints : 3,4

C) prints : 5,6

D) causes compile errors.

15) Inspect the following code:

```
#include <stdio.h>
struct point{
    int x;
    int y;
};
void main(void){
    struct point p1={1,2};
    struct point p2={3,4},p3={5,6};
    struct point *ptr;
    p1=p3;
    ptr=p2;
    printf("%d,%d\n",(*ptr).x, (*ptr).y);
}
```

This code :

- A) prints : 1,2
- B) prints : 3,4
- C) prints : 5,6
- D) causes compile errors.

16) Inspect the following code:

```
#include <stdio.h>
void main(void){
    {
        int i=4;
        printf("i=%d, ",i);
    }
    printf("i=%d\n",i);
    int i=7;
}
```

This code:

- A) prints i=4,
- B) prints i=4, i=7
- C) prints i=4, i=4
- D) causes compile errors.

17) Inspect the following code:

```
#include <stdio.h>
#include <stdlib.h>
struct point{
    int x;
    int y;
};
void main(void){
    struct point p1={1,2};
    struct point p2={3,4},p3={5,6};
    struct point *ptr;
    ptr=(struct point *)malloc(sizeof(struct point));
    (*ptr)=p2;
    printf("%d,%d\n",(*ptr).x, (*ptr).y);
}
```

This code :

- A) prints : 1,2
- B) prints : 3,4
- C) prints : 5,6
- D) causes compile errors.

18) Inspect the following code:

```
#include <stdio.h>
int i=7;
void main(void){
    {
        i=i+1;
        int i=4;
        printf("i=%d, ",i);
    }
    printf("i=%d\n",i);
}
```

This code:

- A) prints i=4, i=7
- B) prints i=5, i=7
- C) prints i=4, i=8
- D) causes compile errors.

19) Inspect the following code:

```
#include <stdio.h>
int i=7;
void main(void){
    {
        int i=4;
        printf("i=%d, ",i);
    }
    printf("i=%d\n",i);
}
```

This code:

- A) prints i=4,
- B) prints i=4, i=7
- C) prints i=4, i=4
- D) causes compile errors.

20) Inspect the following code:

```
#include <stdio.h>
void main(void){
    int i=4;
    printf("i=%d, ",i);
    int i=7;
    printf("i=%d\n",i);
}
```

This code:

- A) prints i=4,
- B) prints i=4, i=7
- C) prints i=4, i=4
- D) causes compile errors.

- end of section-

Section 2 **(20 marks total, each program receives different marks.)**

(if you need additional space, please write the code on a separate paper, write your EN# and attach that paper to this paper)

- 1) Write a small program to print all odd numbers from 1 to 100. You should show all the code needed for this program, including any libraries. **(4 Marks)**

2) Write a program to sort N numbers stored in an array, in ascending order . Assume the following code is already provided. Start writing your code in main() below. **(8 Marks)**

```
#include <stdio.h>
```

```
int sort[]={23, 11, 56 , 2, 7, 21, 3, 1};
```

```
int count = 8;    // The number of values in the array to sort.
```

```
int main(void){
```

3) Write a program to copy values in one type of structure to another. Use the definitions listed below. (use the provided code as the start of your program) **(8 Marks)**

```
#include <stdio.h>
```

```
struct cord_3d{  
    int x;  
    int y;  
    int z;  
};
```

```
struct cord_2d{  
    int x;  
    int y;  
};
```

```
int main(void){  
    // assume cord_3d has been initialized as below:  
    struct cord_3d p3_1={1,1,1};  
    struct cord_3d p3_2={2,2,2};  
    struct cord_3d p3_3={3,3,3};  
  
    // now you need to write the code to allocate 3 variables for 'struct cord_2d'  
    // and copy the x and y values in p3_1, p3_2, p3_3 to your 3 new variables.  
    // finally write the code to print the values in each of the 3 variables you allocated.  
    // your out put should look like (1,1) followed by (2,2) followed by (3,3)
```

- end of Examination -