Shri Vile Parle Kelavani Mandal's

SHRI BHAGUBHAI MAFATLAL POLYTECHNIC

Computer Engineering Department

Control Structure Part - IV

Course: Programming in C

Course Code: PRC238912

Name of Staff: Mr. Pratik H. Shah

SEMESTER : II

DIVISION : A

Course Outcome – 2

Conceptualize loops and control structure

Student will be able to

- Use break and continue statement with loops and switch case.
- Differentiate break and continue statement

Break Statement

1. Keyword
2. It is used to bring the program control out of the loop.
3. It is used inside loops or switch statement.
4. It breaks the loop one by one

Syntax: Break; // in case of loop or switch

Break Statement with for loop

```
#include<stdio.h>
void main ()
{
   int i;
   for(i = 0; i<10; i++)
   {
      printf(“%d\n",i);
      if(i == 5)
      break;
   }
   printf("came outside of loop i = %d",i);
}
```

Output

```
0
1
2
3
4
5
came outside of loop i=5
```

Break Statement with while loop

```
#include<stdio.h>
void main ()
{
   int i = 0;
   while(i<20)
   {
      printf("%d  ",i);
      i++;
      if(i == 10)
      break;
   }
   printf("came out of while loop");
}
```

Output

```
0  1  2  3  4  5  6  7  8  9  came out of while loop
```

Break Statement with do-while loop

```
#include <stdio.h>
void main ()
{
    int a = 10;
   do
   {
       if( a == 15)
      {
         a = a + 1;
         break;
      }
      printf("value of a: %d\n", a);
      a++;
   } while( a < 20 );
}
```

Output

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
```

Break Statement with Switch Case

```
#include<stdio.h>
void main()
{
int number;
printf("enter a number:");
scanf("%d",&number);
switch(number)
        {
        case 1:printf(“You have entered 1");
        break;
        case 2:printf(" You have entered 2");
        break;
        case 3:printf(" You have entered 3");
        break;
        default:printf("number is not equal to 1,2 and 3");
        }
}
```

Output

First time Execution:

```
enter a number:2
You have entered 2
```

Second time Execution:

```
enter a number:5
number is not equal to 1,2 and 3
```

Continue Statement

Continue Statement

1. Keyword
2. It is used to bring the program control to the beginning of the loop.
3. skips some lines of code inside the loop and continues with the next iteration.

Syntax: Continue; // in case of loop

Continue Statement with for loop

```
#include<stdio.h>
void main()
{
      int i;
     for(i=1;i<=10;i++)
    {
        if(i==5)
        {
           continue;
        }
        printf("%d \n",i);
    }
}
```

Output

```
1
2
3
4
6
7
8
9
10
```

Continue Statement with while loop

```
#include <stdio.h>
void main()
 {
    int i = 0;
    while (i < 10)
    {
        if (i == 4)
       {
            i++;
           continue;
       }
       printf("%d\n", i);
       i++;
    }
 }
```

Output

```
0
1
2
3
5
6
7
8
9
```

Continue Statement with do-while loop

```
#include <stdio.h>
void main ()
{
    int a = 10;
   do
   {
       if( a == 15)
      {
         a = a + 1;
         continue;
      }
      printf("value of a: %d\n", a);
      a++;
   } while( a < 20 );
}
```

Output

```
value of a: 10
value of a: 11
value of a: 12
value of a: 13
value of a: 14
value of a: 16
value of a: 17
value of a: 18
value of a: 19
```

Break vs Continue

Break

1. This statement allows a user to exit a loop's overall structure.
2. It can be written as:
   break;
3. We can use break statement with switch and loops.
4. 'break' resumes the control of the program to the end of loop enclosing that 'break'.

Continue

1. It does not allow a user to exit an overall loop structure.
2. It can be written as:
   continue;
3. We can use continue statement with loops but not with switch.
4. 'continue' resumes the control of the program to the next iteration of that loop enclosing 'continue'.

Conclusion

Both the break and continue statements are jump statements that move control to another section of the programme.

Whereas the break statement allowed the control to exit the loop, the continue statement allowed the control to enter the loop's next iteration.

Exercise

1. What will be the output of the following code:

```
#include <stdio.h>
void main()
{
  int outside,inside;
  for(outside=0;outside<=3;outside++)
  {
      for(inside=0;inside<=3;inside++)
      {
          printf("outside=%d & inside = %d \n",outside,inside);
         if(outside==2 && inside==1)
         {
             break;
         }
      }
   printf("\n");
  }
}
```

Correct Answer

```
outside=0 & inside = 0
outside=0 & inside = 1
outside=0 & inside = 2
outside=0 & inside = 3

outside=1 & inside = 0
outside=1 & inside = 1
outside=1 & inside = 2
outside=1 & inside = 3

outside=2 & inside = 0
outside=2 & inside = 1

outside=3 & inside = 0
outside=3 & inside = 1
outside=3 & inside = 2
outside=3 & inside = 3
```

2. What will be the output of the following code:

```
#include <stdio.h>
void main()
{
    int i;
    for(i=0;i<20;i++)
   {
        if(i%2==0)
       {
            continue;
       }
       printf("%d ",i);
   }
}
```

Correct Answer

1 3 5 7 9 11 13 15 17 19

3. What will be the output of the following code:

```
#include <stdio.h>
void main()
{
    int a = 10;
    do
    {
         if( a == 15)
        {
            a = a + 5;
            continue;
        }
        printf("%d ", a);
        a++;
    } while( a < 30 );
}
```

Correct Answer

10 11 12 13 14 20 21 22 23 24 25 26 27 28 29

4. What ‘C’ program to print following patterns using nested for loop and break statement

1

1 2

1 2 3

1 2 3 4

Correct Answer

```
#include <stdio.h>
void main()
{
        for (int i = 1; i <= 6; i++)
       {
            for (int j = 1; j <= i; j++)
            {
                if (i <= 4)
                {
                    printf("%d ", j);
                }
               else
               {
                   break;
               }
            }
            printf("\n");
       }
}
```

5. Write ‘C’ program to calculate sum of 10 numbers (max 10 numbers) and if the users enters negative number then loop should be terminated

Correct Answer

```
#include <stdio.h>
void main()
{
      int i,number,sum=0;
     for (i = 1; i <= 10; i++)
    {
          printf("Enter n%d: ", i);
          scanf("%d", &number);
          if (number < 0)
         {
               break;
          }
          sum += number;
    }
    printf("Sum = %d", sum);
}
```

Conclusion

Will you be able to now use break and continue statement with loops and switch case?

Can you identify any 2 points for difference between break and continue?

THANK YOU !!!!